Univerzita Karlova

Přírodovědecká fakulta

Studijní program: Geografie (bakalářské studium)

Studijní obor: Geografie a kartografie

Vít Kovačka

DESAGREGACE PROSTOROVÝCH DAT O HUSTOTĚ ZALIDNĚNÍ S VYUŽITÍM 3D MODELU MĚSTA

SPATIAL DISAGGREGATION OF POPULATION DATA USING 3D CITY MODEL

Bakalářská práce

Praha 2020

Vedoucí bakalářské práce: Mgr. Lukáš Brůha, Ph.D.

Prohlášení o autorství

Prohlašuji, že jsem závěrečnou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje a literaturu. Tato práce ani její podstatná část nebyla předložena k získání jiného nebo stejného akademického titulu.

Jsem si vědom toho, že v případě použití výsledků získaných v této práci mimo Univerzitu Karlovu je možné pouze po písemném souhlasu této univerzity. Svoluji k zapůjčení této práce pro studijní účely a souhlasím s tím, aby byla řádně vedena v evidenci vypůjčovatelů.

V Praze, dne 02. 5. 2020

Vít Kovačka

Poděkování

Tímto bych rád poděkoval svému školiteli Mgr. Lukáši Brůhovi, Ph.D. za pomoc při výběru tématu, cenné rady, a především ochotu a trpělivost. Mé díky rovněž patří Mgr. Janu Šimberovi, jehož práce zajistila potřebný aplikační rámec pro tuto bakalářskou práci a který trpělivě zodpovídal všechny mé dotazy. Rovněž děkuji Mgr. Janu Sýkorovi za poskytnutá data. Na závěr bych rád vyjádřil svůj vděk rodině, jež mě podporovala během celého studia.

Abstrakt

Práce se zabývá využitím existujících 3D modelů jako zdroje informací o objemu budov, jež je dále využíván ve statistickém modelování prostorových dat. Představeny jsou existující přístupy k využití prostorových informací v modelování a desagregaci, včetně využití trojdimenzionálních dat. Metoda získávání informace o objemu je implementována v prostředí ArcPy pro formát multipatch a v open source prostředí PostgreSQL PostGIS pro rastry obsahující informace o výšce zástavby. Desagregace byla provedena s 2D i s 3D daty a je hodnocena z hlediska přesnosti, výkonu modelu a schopnosti 3D dat nahradit některá 2D data. Navržené metody výpočtu i výsledky modelu jsou kriticky zhodnoceny.

Klíčová slova: 3D data, objem, Praha, desagregace, modelování, ArcPy, PostGIS, strojové učení

Abstract

The thesis studies the use of existing 3D models as a source of information about the volume of buildings, which is further used in statistical modeling of spatial data. Existing approaches to the spatial data disaggregation are presented, including those utilizing three-dimensional data. The method of obtaining volume information is implemented employing ArcPy libraries for multipatch format. Open source PostgreSQL PostGIS database functions were put in use to retrieve the volumes from rasters containing information about the height of the building. Disaggregation, performed with both 2D and 3D data, is evaluated in terms of accuracy, model performance, and the ability of 3D data to replace some 2D data. The proposed calculation methods and model results are critically evaluated.

Keywords: 3D data, volume, Prague, disaggregation, modeling, ArcPy, PostGIS, machine learning

Obsah

Seznam tabulek

Seznam obrázků

1 Úvod

Význam 3D dat v oblasti vizualizace je nesporně velký, neboť lidská představivost snáze uchopí pro ni přirozený třetí rozměr, bez nutnosti prostorové imaginace z druhé dimenze. S 3D modely a vizualizacemi se setkáváme již běžně, ať jde o budoucí developerské projekty, rekonstrukce dávno zaniklých míst, či rozvíjející se 3D tisk. Mimo vizuální přitažlivosti 3D modelů můžeme také uvažovat o jejich využití v rámci různých analýz, kde informace o tvaru objektů, obsažená v 3D modelu může být hodnotným zdrojem dat pro další zpracování.

A právě funkce 3D modelu v prostorové analýze tvoří podstatu tématu této práce. Základní myšlenkou tématu je využití 3D modelu města Prahy k výpočtu objemů jednotlivých budov. Nově získaná informace o objemu je následně zahrnuta do statistického modelu, vytvořeného Janem Šimberou (2020). Původní model desagreguje prostorová data na základě strojového učení na 2D datech.

Cílem práce je zjistit, zdali nově přidaná informace, tedy objem budov, zpřesní výstup z modelu. Cílem zkoumání budou i další faktory, jako je změna výkonu modelu s rozšířením vstupních dat modelu o data objemová. Dále bude ověřena schopnost modelu učení se na menším vzorku dat. A nakonec také nahraditelnost některých 2D datových sad právě 3D daty, což může být užitečné při celostátní studii, která bude zahrnovat území, pro které některé 2D datové vstupy nejsou k dispozici.

Dílčími cíli práce bude seznámit se s existujícími postupy v oblasti desagregace a dostupnými 3D daty. Bude provedena rešerše modelu Jana Šimbery (2020), příprava aplikačního prostředí a testování modelu s cílem porozumění jeho kódu, posloupnosti a principu běhu. Na tomto základě bude navržen a implementován postup automatického

zpracování 3D dat tak, aby byla kompatibilním vstupem pro model. Po vybrání vhodných trénovacích území bude provedeno modelování a desagregace populačních hodnot s různou konfigurací modelu, tedy se zahrnutím různých dat do modelování. Výsledky desagregace s různým nastavením modelu budou zhodnoceny na základě skutečných hodnot a vizualizovány.

Bakalářská práce má následující strukturu. V rešeršní části je podrobněji popsán princip desagregace, včetně již existujících využití 3D dat při desagregaci. Rozebírány jsou i další teoretické aspekty ovlivňující empirickou část této práce, jako dasymetrické mapování či datové reprezentace. Aplikační rámec, tedy statistický model Jana Šimbery (2020), programovací a databázové prostředí popisuje následující kapitola. V metodické části je popsán způsob úpravy dostupných 3D dat, jejich implementace do modelu i chod modelu samotného, včetně validace výsledků. Výsledky jsou diskutovány a vizualizovány v posledních dvou kapitolách.

2 Úvod do problematiky

2.1 Desagregace prostorových dat

V demografii hojně využívané geografické informační systémy (GIS) nabízejí řadu možností analýzy, mapování či vizualizace populačních dat. Jednou z oblastí využití GIS je geostatistické modelování, jehož cílem je odhadnout populaci územního celku v případě nekompletních dat ze sčítacích šetření, či v případě potřeby dat za jemnější územní celky, než jsou existující data (Biljecki et al. 2016). Šimbera (2016) uvádí třetí výhodu desagregovaných dat, a to že jsou anonymizovaná a z části náhodná, nepodléhají tedy ochraně osobních údajů a mohou být volně používána.

2.1.1 Definice prostorové desagregace

Goodchild a Lam (1980, cit. v Šimbera 2016, s. 13) definuje prostorovou desagregaci jako jednu z disciplín prostorové interpolace, která na základě známých hodnot sledované charakteristiky pro určité územní celky vytváří tyto hodnoty pro celky jiné, kde cílový územní celek bývá menší než zdrojové území. Šimbera (2020) uvádí následující rovnici, kde *M* představuje sledovanou hodnotu pro celý územní celek, *i* jsou cílové menší územní jednotky a w_i relativní váhy desagregace.

$$\hat{m}_i = M \frac{w_i}{\sum_i w_i}$$

2.2 Dasymetrické mapování

Základní přístup k desagregaci je založen na ploše prvků, kde prostorové vymezení sledovaného jevu je hlavní vahou (Sadahiro 1999). Šimbera (2016) uvádí, že více sofistikovanou metodou, zvanou dasymetrické mapování, je využití dodatečných ukazatelů pro výpočet relativních vah. Jako důvod uvádí možnost využití desagregace mimo prostorovou vědu, například v rámci inženýrství elektrické sítě, kde je využíváno strojové učení spotřeby elektrické energie domácnostmi v rámci distribuční sítě.

Dasymetrické mapování či dasymetrické modelování označuje techniku, kde dodatečná data, například land use, land cover, digitální model terénu (DMT), délka ulic apod., slouží jako vodítko k nalezení populačních hodnot pro data s větším prostorovým rozlišením (Bakilah et al. 2014).

- ***Binární dasymetrické mapování*** je metodou, při které jsou zdrojová území rozdělena na osídlené a neosídlené plošky. Relativní váhy jsou posléze aplikovány mimo území osídlených zdrojových plošek. Nevýhodou této metody je předpoklad, že odlehlé oblasti jsou osídleny rovnoměrně (Maantay, Maroko, Herrmann 2013).
- ***Multi-class dasymetrické mapování*** částečně řeší tento problém tím, že zdrojové území je rozděleno do tříd z hlediska hustoty zalidnění. K odhadu hustoty zalidnění se poté používá vícerozměrná lineární regrese s metodou nejmenších čtverců (Langford 2006).

Pro výše zmíněné přístupy je typické podcenění odhadu sledované charakteristiky (hustoty zalidnění) v urbánních oblastech a přecenění v oblastech odlehlých (Šimbera 2020). Jiné dva základní typy dasymetrického mapování uvádí Li a Zhou (2018).

- ***Klasifikované váhy*** zahrnují obě výše zmíněné metody dasymetrického mapování, tedy binární a multi-class. Zvýšením počtu tříd se sice zvýší přesnost (Dmowska, Stepinski 2017), nicméně stále nelze zachytit variace populace v rámci každé třídy.
- ***Dasymetrické mapování založené na kontinuálních vahách*** vzniklo v návaznosti na zlepšující se dostupnost a kvalitu doplňujících dat, související s rozvojem družicové techniky, LIDAR technologií, zpřesnění DMT a všeobecně vyšší dostupností prostorových dat. Algoritmy dasymetrického mapování jsou rovněž náročné na výpočetní výkon, jejich vývoj v minulých letech ovlivnila

vyšší výpočetní kapacita počítačů (Li, Zhou 2018). Metoda dasymetrického mapování s kontinuálními váhami počítá s dynamickým generováním vah z různorodých prostorových dat, kde je zkoumán statistický vztah mezi sledovanou populační charakteristikou a množstvím prostorových proměnných.

V posledních padesáti letech byla pro dasymetrické mapování využívána zejména 2D data, tedy satelitní snímky a GIS data (Biljecki et al. 2016). Například Bakilah et al. (2014) či Langford et al. (2008) využívají dostupná prostorová big data, jako body zájmu, restaurace, síť silnic a železnic etc. Sutton (1997) a Anderson et al. (2010) odhadují populaci na základě nočních leteckých snímků s předpokladem, že existuje vztah mezi světelným znečištěním a počtem obyvatel. Jak ovšem píše Wang et al. (2018), některé zdroje nočních snímků mají problém s přesvícením, v praxi je tedy užitečné kombinovat noční snímky s dalšími datovými zdroji. Pozzi a Small (2005) odhadují počet obyvatel na základě vegetačního pokryvu, kde menší množství vegetace indikuje větší počet obyvatel.

Zajímavou skupinou zdrojových dat jsou georeferencovaná data ze sociálních sítí a mobilních služeb. Zasina (2018) využívá API Instagramu a za pomoci informace o poloze, obsažené v metadatech fotografií, odhaduje rozložení obyvatelstva v centru města Lodž. Steiger et al. (2015) analyzují georeferencované tweety na sociální síti Twitter, kde za pomocí klíčových slov rozlišují domácí a pracovní sociální aktivity k odhadu mobility a populačních dat.

2.2.1 Footprinty budov

Většina výše zmíněných prací operovala s footprinty budov, tedy kolmým průmětem budovy na zemský povrch. Nejjednodušším přístupem je vycházet z celkového počtu budov ve sledovaném území či z jejich plochy. Například Wu, Wang a Qiu (2008) ve své práci desagregují volně dostupná populační data na úrovni bloků v Austinu na základě footprintů budov. Biljecki et al. (2016) ovšem poukazují, že tato metoda dosahuje uspokojivých výsledků v relativně homogenní oblasti, nicméně v oblastech s vyšší variabilitou počtu pater jednotlivých budov dochází k chybám.

2.2.2 Implementace 3D dat

Výše zmíněnou výškovou a objemovou variaci budov řeší využití třetího rozměru. S rozvojem dálkového sběru dat, jako je letecká fotogrammetrie či LIDAR je nyní možné

automaticky měřit výšky budov, s jejichž pomocí může být vytvořena 3D objemová reprezentace jednotlivých objektů (Biljecki et al. 2016). Lu, Im a Quackenbush (2011) na příkladu části Denveru porovnávají výsledky desagregace nad footprinty budov a nad lidarovými daty o objemu budov. Při použití regresního modelu na objemová data získávají lepší výsledky než při použití footprintů budov. Naproti tomu Dong, Ramesh a Nepali (2010) při použití lidarových dat s přesností 3-5 metrů při desagregaci části města Denton zaznamenávají signifikantní chyby ve výsledcích. Chyba vzniká již v předzpracování dat, při odlišení budov od vegetace. Zdůrazňují proto nutnost použití detailních lidarových dat.

Vzhledem k enormnímu objemu dat a jejich nejednotnosti či nedostupnosti pro větší území se většina studií zaměřuje na případovou studii města či menšího území. V tomto ohledu je zajímavá práce Biljeckiho et al. (2016), která zpracovává odhad populace na základě 3D modelu celého Nizozemska. Autoři rovněž poukazují na zvyšující se dostupnost 3D dat, která podle nich v následujících letech již nebudou výsadou bohatých zemí.

2.3 LOD

Jak poukazuje Biljecki et al. (2016), můžeme rozlišovat dvě roviny úrovně detailu, tedy level of detail (LOD), a to úroveň geometrickou a sémantickou (viz obrázek č. 1). Odlišení úrovně detailu je důležité pro porovnání výsledků a zhodnocení užité metody, neboť se zvyšující se LOD by se teoreticky měly zpřesňovat výsledky desagregace.

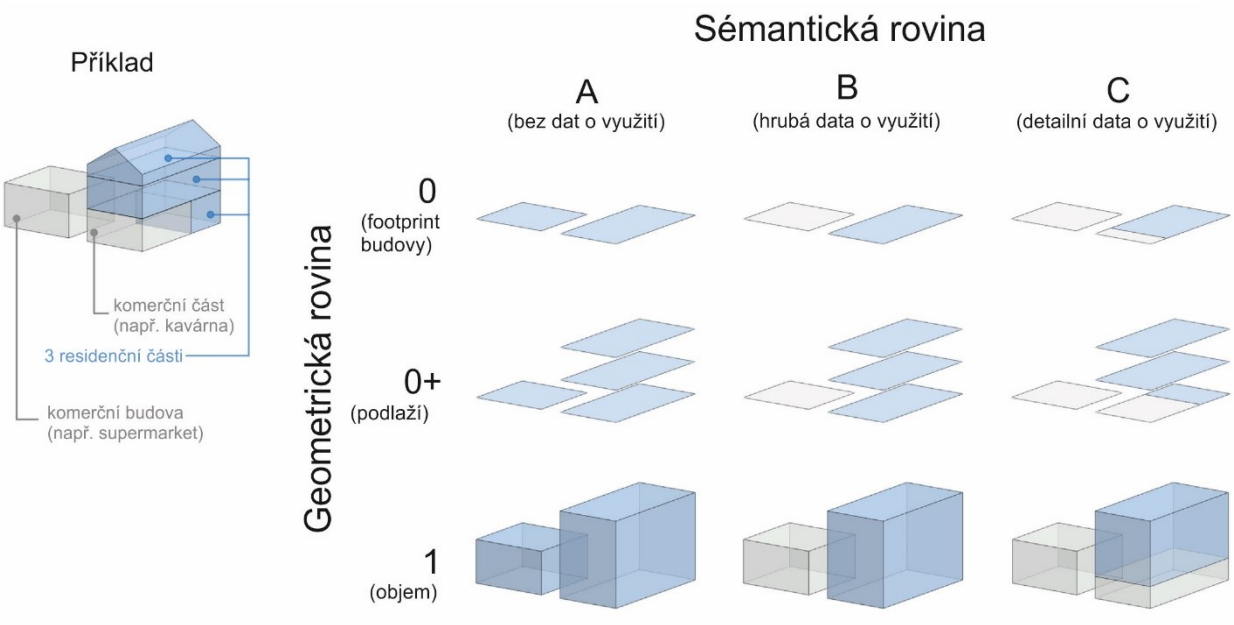
2.3.1 Rovina geometrická

Z hlediska geometrického můžeme rozlišovat tři úrovně detailu. Pro základní desagregaci je využíváno footprintů budov, tedy 2D dat (úroveň 0). Úroveň 0+ označuje rovněž 2D dataset, udávající informaci o počtu pater. Taková data jsou z geometrického hlediska stále 2D, nicméně již nesou 3D informaci a při odhadu průměrné výšky patra mohou sloužit k hrubému výpočtu objemu budovy. Pokud zdi budovy nejsou po celém obvodu svislé, plocha pater upřesní celkovou plochu podlah dané budovy. Poslední úroveň geometrické roviny je úroveň 1, tedy plnohodnotná 3D datová reprezentace budovy.

2.3.2 Rovina sémantická

Druhá rovina určuje využití budovy. Je neméně důležitá, neboť rozlišení residenčních budov a budov industriálních, obslužných etc., jako jsou továrny, nádraží, skladiště, muzea, památníky či obchodní centra je klíčové k přesné desagregaci. Jiné budovy než residenční mají obvykle velkou plochu i objem. Úroveň A označuje data bez informace o využití budovy. B označuje data s informací za celý footprint, tedy pokud se jedná například o typický městský partér, kde v přízemí probíhá komerční činnost, ale vyšší patra plní residenční funkci, mohou být informace za celý footprint zkreslené. Tento problém řeší úroveň C, kde je footprint, případně patro či 3D model rozdělen na residenční a komerční část.

Obrázek č. 1: Sémantická a geometrická rovina LOD, převzato a upraveno (přeloženo) z Biljecki, Ohori, Ledoux, Peters a Stoter (2016). Modré plošky a objekty představují residenční část budovy, bílé část komerční.



2.4 Datové reprezentace

Reprezentace 3D těles je obsáhlé a vyvíjející se téma. Vhledem k aplikačnímu zarámování této práce bude přiblíženo z hlediska geoinformatiky a reprezentací využitých v následujících kapitolách. Všechny popsané nástroje, pokud není řečeno jinak, se vztahují k softwaru ArcMap 10.5.3. Pro obecnější přehled doporučuji knihu Moderní počítačová grafika (Žára et al. 2005).

2.4.1 Surface data

V rámci GIS prostředí se jedná o data udávající výšku od referenčního tělesa, tedy elipsoidu či koule, či od jiné referenční roviny. Data mohou být členěna na buňky (rastr) či na trojúhelníkovou síť plošek. Při použití tohoto typu dat často dochází k záměně s tzv. 2,5D daty. Ta jsou definována tak, že pro každou x a y souřadnici existuje právě jedna z souřadnice. Data o nadmořské výšce tedy například udávají právě jednu hodnotu z v unikátním místě, což znemožňuje vytvoření jakýchkoliv převisů, jeskyní či kolmých stěn. Příkladem může být DMT, digitální model reliéfu (DMR), relativní výšky budov etc. Souřadnice z je zde udávána formou atributu.

2.4.2 3D feature data

Feature data jsou samostatné objekty, které mají 3D informace uložené ve své geometrii. Trojrozměrná data mohou obsahovat více z souřadnic v jednom x, y bodě. Na příkladu budovy mohou být stěny svislé, základy, podlaha v prvním patře a střecha mohou mít rozdílnou z souřadnici a stejnou x, y souřadnici (Esri 2020a).

2.4.3 Multipatch

Standard, vyvinutý společností Esri v roce 1997. Slouží k ukládání složitějších 3D objektů, jako jsou budovy, geologické celky či bezletové zóny, a to buď v geodatabázi, nebo ve formátu shapefile. Samotná geometrie multipatche se skládá z části plošné a řídící, jenž určuje pořadí a interpolaci vrcholů. Multipatch může vzniknout třemi způsoby:

- Speciálními geoprocesingovými nástroji v toolboxu 3D Analyst, jako je Extrude Between či Layer 3D to Feature Class.
- Importem existujících modelů formátu třetí strany, jako je SketchUp, COLLADA či 3D Studio Max.
- ArcObjects skriptováním.

Výhodou formátu multipatch je jeho schopnost obsahovat textury či být vykreslován průhledně (Esri 2020b).

2.4.4 Objemový model

Pro výpočet objemu tělesa, v tomto případě budovy ohraničené zespodu terénem, ze stran obvodovými stěnami a shora střechou je nutné, aby těleso bylo uzavřené, tedy

neobsahovalo mezery ani díry mezi jednotlivými ploškami geometrie. Zdali multipatch tvoří objemový model lze ověřit pomocí nástroje Is Closed 3D v toolboxu 3D Analyst, jehož výstupem je True v případě, že multipatch tvoří objemový model a False v případě opačném. Pokud objekt uzavřený není, je možné na něj aplikovat nástroj Enclose Multipatch, který bude detailněji popsán v následujících kapitolách.

2.4.5 WKT

Well-known text je textový značkovací jazyk pro reprezentaci vektorové geometrie. Jeho ekvivalentem je well-known binary (WKB), který ukládá geometrii v binárním kódu. Oba formáty definovalo Open Geospatial Consortium (OGC) a popsalo v Simple Feature Acces (Herring 2011). V zásadě jde o relativně lehce čitelnou textovou reprezentaci geometrie, která podporuje základní vektorové objekty jako jsou body, linie a polygony, ale i 3D datové reprezentace Triangulated irregular network (TIN) či PolyhedralSurface. Vzhledem ke stále rozšiřujícím se možnostem Open GIS a volně dostupného geoinformačního a databázového softwaru je nutné uvést WKT jako jednu z možných budoucností 3D modelování.

3 Aplikační rámec

Aplikačním a z velké části i teoretickým východiskem této práce je model Jana Šimbery, a to sice Reconfigurable Urban Modeler (RUM). Jedná se o nástroj, který umožňuje desagregaci prostorových dat na základě strojového učení. V rámci diplomové práce Jana Šimbery (Šimbera 2016) byl RUM představen jako toolbox pro ArcGIS. Diplomová práce je přínosná zejména díky detailnímu popisu dostupných prostorových dat, použitelných při dasymetrickém mapování.

V rámci další práce Jana Šimbery, jejímž výstupem je mimo jiné článek otištěný v časopise Cartography and Geographic Information Science (Šimbera 2020), vznikla druhá verze RUMu. S touto verzí modelu pracuje má práce. Model je volně dostupný na GitHubu na odkaze: *https://github.com/simberaj/rum*

3.1 Reconfigurable Urban Modeler

RUM je popsán v readme na výše zmiňovaném GitHubu. Tvoří ho soustava skriptů v Pythonu 3.6+. Skripty pracují s několika externími moduly, které mohou být instalovány příkazem `pip install -r requirements.txt`, kde `requirements.txt` je textový soubor, ve kterém je uložen seznam potřebných modulů. Skripty pracují s databází PostgreSQL 10 a vyšší. Databáze musí mít nainstalované rozšíření PostGIS. Přístup skriptů do databáze je modifikován textovým souborem ve formátu JSON.

Samotné spouštění skriptů se předpokládá skrze příkazovou řádku systému Windows, kde je využit modul `argparser`, zajišťující parsování argumentů v příkazové řádce a v rámci skriptu. Každý skript je krátce popsán a jsou vypsány jeho povinné a volitelné argumenty.

3.1.1 Schéma RUMu

Popis jednotlivých skriptů je uveřejněn ve zmiňovaném readme, stejně jako v docstringu každého skriptu, není nutno ho tedy znovu uvádět. Nutné je ovšem přiblížit posloupnost, v jaké skripty pracují.

Inicializace – v prvotní fázi je vytvořeno databázové schéma, ve kterém následující skripty pracují. Po nahrání první vrstvy prostorových dat do schématu je vytvořen extent, tedy vrstva ohraničující zájmové území. Tento extent je překryt pravoúhlou mřížkou o velikosti jedné buňky 100x100 metrů.

Import dat – v druhé fázi jsou do databázového schématu nahrána všechna data, která budou použita při následujícím trénování modelu a aplikaci modelu. Data mohou být v jakémkoliv formátu dle normy GDAL/OGR/Fiona. Součástí je i import dat z OpenStreetMaps (OSM). Této kapitole se detailně věnuje diplomová práce Jana Šimbery (Šimbera 2016).

Úprava dat – po nahrání dat do databáze je možné je dodatečně upravit, například provést rekategorizaci atributů či vypočítat doplňující charakteristiky pro polygonové vrstvy.

Výpočet vstupních vrstev – z dat, jež byla nahrána a upravena, jsou pomocí konfiguračního souboru ve formátu JSON vypočteny nové vrstvy s předponou `feat_`. Tyto vrstvy jsou spojeny s mřížkou pomocí atributu `geohash`, což je unikátní kód pro každý čtverec mřížky. Nutné je také vypočítat `target`, tedy proměnnou, jež má být desagregací získána, na část trénovacích dat.

Trénování modelu – před trénováním modelu je nutné spojit všechny vstupní vrstvy do jedné tabulky. Následně je model natrénován. K dispozici je osm typů modelů skrze modul `scikit-learn`. Trénováním modelu je vytvořena tabulka vah pro aplikaci modelu.

Aplikace modelu – při aplikaci modelu je na základě tabulky vah pro každou buňku mřížky odhadnuta hodnota sledované proměnné `target`.

Desagregace – na základě odhadnuté hodnoty je desagregována hodnota stejného jevu z území většího, než jsou buňky mřížky.

Validace – v poslední části je přesnost odhadnutých hodnot statisticky hodnocena dle hodnot skutečných. O validaci více v metodické části.

3.1.2 Modelování

RUM je koncipován tak, že trénování modelu probíhá v jiném databázovém schématu než aplikace modelu. Model je ukládán jako samostatný soubor na disku. To umožňuje trénování modelu na jiném území, tedy i na jiném městu, než je posléze aplikován. V praxi takto Jan Šimbera trénoval model na Praze a jeho aplikaci následně provedl i na Maribor a Tallin. Předpokládá se podobná struktura města, proto například aplikace na americká města, stavěná s odlišnou architektonickou koncepcí, nemusí být natolik přesná, jako na podobná evropská města.

3.2 PostGIS

Jak bylo řečeno, RUM pracuje s daty uloženými v objektově relační PostgreSQL databázi. Výhodou této databáze je MIT licence, díky které spadá do kategorie open source. Prostorové dotazy nad PostgreSQL databází umožňuje rozšíření PostGIS, distribuované pod GNU GPL licencí. Po instalaci PostGIS inicializujeme pro konkrétní databázi příkazem `CREATE EXTENSION postgis;`. K PostGISu existuje celá řada rozšíření, umožňujících širokou škálu prostorových dotazů, operací a analýz, což dělá z PostGISu zajímavý open source GIS nástroj. S Pythonem 3 komunikuje PostgreSQL například skrze modul `psycopg2`, který bude využíván v této práci.

4 Data

V první řadě je nutné vymezit zájmové území. V rámci této práce jej tvoří hlavní město Praha. Dasymetrické mapování vyžaduje velké množství dat o zájmové oblasti, neboť s vyšším množstvím (kvalitních) dat se mapování zpřesňuje. V rámci RUMu je několik skriptů pro nahrání dat. Využívány budou tři a to sice:

`import_layer.py` pro nahrání vektorové vrstvy

`import_raster.py` pro nahrání rastru, který je v databázi uložen jako polygonová vrstva s polygonem pro každou buňku

`import_osm.py` pro nahrání OSM dat, buď komprimovaných ve formátu .bz2, nebo dekomprimovaných

4.1 2D data

Páteří dasymetrického mapování jsou přirozeně 2D data, která tvoří v současnosti nejdostupnější datový zdroj. Níže popsané 2D datové zdroje byly využity při modelování v rámci této práce:

- ***OpenStreetMap*** – data OSM, stažená skrze server *https://download.geofabrik.de* ke dni 20.1.2020. (GEOFABRIK 2020).
- ***Urban Atlas 2012*** – pro Prahu (Evropská komise 2012).
- ***SRTM 1 Arc-Second Global*** – digitální model terénu (USGS 2020).
- ***ArcČR 500 ver. 3.3*** – vrstva území hlavního města Prahy (ARCDATA 2020).
- ***Budovy s číslem domovním a vchody*** – bodová vrstva obsahující adresní body pro každou budovu v Praze, kde zájmovým atributem je BUDOBYOSL, tedy počet obyvatel v budově dle sčítání lidu, domů a bytů (SLDB) 2011 – obvyklý pobyt vč. cizinců (SLDB 2011).

4.1.1 Hodnocení 2D dat

V případě OSM můžeme pro Prahu předpokládat vysokou přesnost dat, zejména díky importům z Registru územní identifikace, adres a nemovitostí (RÚIAN) a z katastru. Nevýhodou může být nemožnost stažení dat pouze pro území Prahy, kde při exportu dat přímo z geoprohlížeče OSM je export limitován na 50 000 entit, což je číslo pro Prahu hluboce nedostačující. Řešením se stalo využití serveru GEOFABRIK, kde jsou nabízena denně aktualizovaná OSM data za různé územní celky. Zde byla stažena data pro celou ČR a následně nahrána do PostgreSQL databáze. Import dat do databáze je operací časově velmi náročnou, v případě komprimovaných dat se jednalo o desítky hodin, v případě dat nekomprimovaných o hodiny, neboť celkový čas nebyl prodloužen dekomprimací dat. Použitím dat pouze pro Prahu by byl tento čas výrazně zkrácen.

Urban Atlas spadá do programu Copernicus a nabízí rastrová data o využití povrchu pro širší území vybraných velkých měst. Přesnost rastru je udávána 5 metrů (Evropská komise 2016), avšak může se lišit v závislosti na sledovaném území.

Dva listy SRTM rastru, konkrétně list SRTM1N49E4V3 a SRTM1N50E014V3, pořízené 23. září 2014, byly využity jako zdroj dat pro DTM, který je dále využíván pro výpočet sklonu svahu a orientace vůči světovým stranám.

ArcČR 500 ver. 3.3 je volně dostupná ESRI geodatabáze, obsahující základní české administrativní jednotky a topografická data. V této práci je využita jako zdroj vrstvy, jež má být desagregována.

Poslední vrstva, a to budovy s číslem domovním a vchody je důležitá pro natrénování modelu. Představuje skutečné hodnoty, které musí model v části území znát tak, aby je mohl pro zbytek území odhadnout. Vzhledem k charakteru informací se jedná o data neveřejná. I přes studijní a výzkumné účely využití dat nebyla zapůjčena ani ze strany IPR, ani ze strany ČSÚ. Data poskytla Urbánní a regionální laboratoř PřF UK.

4.1.2 Limity dat

Mimo výše zmíněných nedostatků představuje hlavní problém rozdílný časový horizont dat. OSM jsou víceméně aktuální, ale Urban Atlas je za rok 2012 a data ze SLDB za rok 2011. Data, jež jsou desagregována, tedy hlavní město Praha z ArcČR 500 ver. 3.3 jsou za rok 2016. To může vést k lokálním nepřesnostem v oblastech nové výstavby,

zejména v blízkých suburbanizačních pásech okolo Prahy, či v případě nové bytové výstavbě ve starých částech Prahy, jako například Residence Garden Towers na Žižkově. Možným řešením by sice bylo použití archivních dat OSM, nicméně 3D data, se kterými především tato práce operuje, jsou dostupná rovněž pouze za aktuální období.

Počet obyvatel, jenž vstupuje do desagregace za celou Prahu je v metadatech popsán jako počet obyvatel na daném území. Není tedy také jasné, zdali stejně jako data s adresními body zahrnuje i cizince, či nikoliv.

Další z problémů zmiňuje Jan Šimbera (2016). Situace, kdy prakticky každý musí mít uvedeno místo svého trvalého bydliště vede k lokálním extrémům v případě radnic, kde jsou hlášeny osoby bez domova.

4.2 3D data

Hůře dostupná, avšak skýtající řadu nových možností, jsou 3D data. Pro tuto práci byla využita následující:

- ***Budovy 3D*** – ve formátu multipatch a shapefileZ (IPR 2019a).
- ***Podlažnosti*** (IPR 2019b).
- ***Relativní výšky budov*** – s velikostí pixelu 1x1 m (IPR 2017).

4.2.1 Hodnocení 3D dat

3D model zástavby města Prahy byl pořizován v letech 2001 až 2008, po poslední aktualizaci odpovídá stavu z roku 2016 (IPR 2020). Model byl zpracováván fotogrammetricky a modelování bylo prováděno v systému MicroStation. Vytvořená data byla následně převedena do formátu shapefileZ a multipatch. Shapefily jsou rozděleny po mapových listech 1 : 5 000. Přesnost modelu je uváděna 1 m.

Formát polygonZ nachází využití ve vizualizaci, jelikož se jedná o 2,5 D formát, který je reprezentován polygony. Nicméně není použitelný pro výpočet sledované charakteristiky, a to sice objemu budovy.

Druhý z ESRI formátů, ve kterých je model distribuován, multipatch, již k prostorovým výpočtům využít lze, nicméně musí splňovat podmínku uzavřenosti, tedy tvořit objemový model. Po ověření nástrojem Is Closed 3D ovšem zjistíme, že v případě 3D modelu zástavby Prahy je uzavřeno okolo 3% všech budov, v závislosti na mapovém listu.

Jako podklad k tvorbě 3D dat můžeme považovat i rastr relativních výšek budov, jež je dostupný s prostorovým rozlišením 1 m. Dostupná jsou i data o počtu podlaží, pomocí kterých můžeme rovněž přibližně aproximovat objem budovy. Výška patra ovšem závisí na architektuře budovy a může být pouze odhadována. Zajímavé by ovšem bylo využití počtu pater spolu s dalšími daty k odhadu například počtu bytových jednotek. Data o podlažnosti také obsahují informaci o využití podkroví, která teoreticky zpřesní vypočtený objem, neboť do modelování by měl vstupovat pouze objem residenční části budovy.

Všechna uvedená data jsou k dispozici pouze v souřadnicovém systému S-JTSK, v Křovákově konformním kuželovém zobrazení.

5 Metodika

Filozofií RUMu bylo vytvořit nástroj, jenž není závislý na platformě a je kompletně open source. Bylo zamýšleno sledovat tuto myšlenku, nicméně vzhledem k charakteru dostupných dat, která jsou ve formátech využívaných softwarem Esri, bylo nutno využít postupy, jež vyžadují aplikace, které nejsou open source. Celkem jsou představeny dva postupy výpočtu objemu skrze modul `arcpy` a jeden open source v rámci PostGIS databáze.

5.1 Úprava 3D dat

Pro výpočet objemu multipatche lze využít nástroje Add Z Information z toolboxu 3D Analyst. Vstupem ovšem musí být uzavřený multipatch. Vzhledem k velkému objemu dat a jejich členění do mapových listů bylo vytvořeno několik skriptů, usnadňujících manipulaci s daty. Psané jsou v Pythonu 2.7, neboť modul `arcpy` pro ArcGIS Desktop neumí pracovat s novější verzí Pythonu. Všechny skripty jsou dostupné zde na GitHubu: *https://github.com/kovackav/3D-GIS-volume*

Pro zadávání argumentů lze využít modul `argparser`, podobně jako u RUMu. Součástí každého skriptu je také krátký docstring. Vzhledem k členění shapefilů dle mapových listů (128 listů) byl využit modul `os`, který vyhledá všechny shapefily v podsložkách cílového adresáře.

`data_calc_shp.py` – vstupním parametrem je adresář, kde jsou data uchovávána. Projde adresář a použije nástroj Enclose Multipatch pro uzavření multipatche s nastaveným parametrem Grid Size 0,05 m. Při větším Grid Size (defaultně je nastaveno 0,15 m) zůstane více budov neuzavřených, nicméně je zkrácen čas výpočtu. Následně je skrze Add Z Information vypočten objem a z multipatche vytvořen footprint. Limitujícím faktorem je výpočetní náročnost, všechny mapové listy byly vypočteny za přibližně třicet hodin. Dalším problémem je také to, že pokud má vstupní multipatch velikost větší než 20 MB, nástroj

Enclose Multipatch nikdy multipatch nedopočítá, ale běží bez jakéhokoliv runtime erroru. V tomto případě je nutné vstupní data manuálně rozdělit do menších souborů.

```
1. import arcpy
2. import os
3. import argparse
4.
5. parser = argparse.ArgumentParser(description = "This script takes
   multipatch files from a folder and encloses them. Then calculates
   volume of the features.")
6. parser.add_argument('-d', '--input_data', type = str, metavar = '',
   required = True, help = 'Folder, where the inputs are stored. It
   can be stored in sub-folders.')
7. args = parser.parse_args()
8.
9. print ('Message: Parameters valid, starting the script.')
10.
11.  arcpy.env.overwriteOutput = 1
12.  data = args.input_data
13.
14.  class LicenseError(Exception):
15.      pass
16.
17.  try:
18.      if arcpy.CheckExtension("3D") == "Available":
19.          arcpy.CheckOutExtension("3D")
20.      else:
21.          raise LicenseError
22.
23.      for root, dirs, files in os.walk(data):
24.          for file in files:
25.              if file.endswith(".shp"):
26.                  inFeature = (os.path.join(root, file))
27.                  outFeature = str(os.path.join(root,
   file)).replace(".shp","") + "_en.shp"
28.                  footprint = str(os.path.join(root,
   file)).replace(".shp","") + "_foot.shp"
29.                  print (inFeature)
30.                  print (outFeature)
31.                  arcpy.EncloseMultiPatch_3d(inFeature, outFeature,
   0.05)
32.                  arcpy.IsClosed3D_3d(outFeature)
33.                  arcpy.AddZInformation_3d(outFeature, 'VOLUME')
34.                  arcpy.MultiPatchFootprint_3d(outFeature,
   footprint)
35.
36.  except LicenseError:
37.      print("3D Analyst license is unavailable")
38.  except arcpy.ExecuteError:
39.      print(arcpy.GetMessages(2))
40.
41.  print ('Message: Script successfully done!')
```

`footprint_volume.py` – vstupním parametrem jsou data, vypočtená předchozím skriptem a rastr relativních výšek budov. Vzhledem k tomu, že některé budovy nejsou nástroj Enclose

Multipatch uzavřeny, musí být dopočítány pomocí rastru relativních výšek budov. V případě Prahy bylo takto dopočteno 193 budov z celkového počtu 228 258 budov.

```
1. import arcpy
2. import os
3. import argparse
4.
5. parser = argparse.ArgumentParser(description = "This script
   calculates volume from footprints of multipatches, which were not
   enclosed and their volume was 0.")
6. parser.add_argument('-d', '--input_data', type = str, metavar = '',
   required = True, help = 'Folder, where the inputs are stored. It
   can be stored in sub-folders.')
7. parser.add_argument('-r', '--input_raster', type = str, metavar =
   '', required = True, help = 'Raster with relative heights of the
   buildings.')
8. parser.add_argument('-o', '--output_folder', type = str, metavar =
   '', required = True, help = 'Folder, where the output should be
   saved.')
9. parser.add_argument('-c', '--condition', type = str, metavar = '',
   required = True, help = 'Optional condition, which determines
   string with which the input ends.')
10.  args = parser.parse_args()
11.
12.  print ('Message: Parameters valid, starting the script.')
13.
14.  arcpy.env.overwriteOutput = 1
15.
16.  foots = []
17.
18.  for root, dirs, files in os.walk(args.input_data):
19.      for file in files:
20.          if file.endswith(args.condition):
21.              print(file)
22.              foot = (os.path.join(root, file))
23.              foots.append(foot)
24.
25.  allfoots = args.output_folder + '/all_foots.shp'
26.  arcpy.Merge_management(foots, allfoots)
27.  outfolder = args.output_folder
28.  z_foots = []
29.  cnt = 0
30.  cnt_z = 0
31.
32.  with arcpy.da.UpdateCursor(allfoots,["FID", "Volume"]) as cursor:
33.      for row in cursor:
34.          cnt = cnt+1
35.          if row[1] == 0:
36.              cnt_z = cnt_z + 1
37.              sql = """{0} =
   {1}""".format(arcpy.AddFieldDelimiters(allfoots, arcpy.Describe(
38.                  allfoots).OIDFieldName), row[0])
39.              arcpy.Select_analysis(in_features=allfoots,
40.
   out_feature_class=os.path.join(outfolder,
   'Shapefile_{0}.shp'.format(row[0])),
41.                                    where_clause=sql)
42.              cursor.deleteRow()
```

```
43.             z_foots.append(os.path.join(outfolder,
   'Shapefile_{0}.shp'.format(row[0])))
44.  all_z_foots = args.output_folder + 'all_z_foots.shp'
45.  relative_heights_raster = args.input_raster
46.  print('Message: ' + str(cnt) + ' buildings were checked, ' +
   str(cnt_z) + ' buildings were calculated.')
47.  class LicenseError(Exception):
48.      pass
49.
50.  try:
51.      if arcpy.CheckExtension("3D") == "Available":
52.          arcpy.CheckOutExtension("3D")
53.      else:
54.          raise LicenseError
55.
56.      arcpy.Merge_management(z_foots, all_z_foots)
57.      arcpy.CreateRandomPoints_management(outfolder, "points",
   constraining_feature_class = all_z_foots,
58.                                          minimum_allowed_distance
   = "0,9")
59.      arcpy.AddSurfaceInformation_3d(outfolder + "\points.shp",
   relative_heights_raster, "Z")
60.      arcpy.Statistics_analysis(outfolder + "\points.shp",
   outfolder + "\points_cal.shp", [["Z", "MEAN"]], case_field = "CID")
61.      arcpy.AddField_management(all_z_foots, "height", "DOUBLE")
62.      arcpy.CalculateField_management(all_z_foots, "height",
   "[Z_Max] - [Z_Min]")
63.      arcpy.AddGeometryAttributes_management(all_z_foots, "AREA")
64.      arcpy.CalculateField_management(all_z_foots, "volume",
   "[height] * [POLY_AREA]")
65.
66.  except LicenseError:
67.      print('3D Analyst license is unavailable')
68.  except arcpy.ExecuteError:
69.      print(arcpy.GetMessages(2))
70.
71.  all_calc = args.output_folder + 'all_calc.shp'
72.  arcpy.Merge_management([allfoots, all_z_foots], all_calc)
73.  arcpy.DeleteField_management(all_calc, ["height", "POLY_AREA",
   "Z_MIN", "Z_MAX"])
74.
75.  del(cursor)
76.  print ('Message: Script successfully done!')
```

`merge_files.py` - skript užitečný pro manipulaci s daty, sloučí všechny shapefily z adresáře do jednoho.

`create_footprint.py` – vstupem jsou multipatche, ze kterých je vytvořen jejich footprint (kolmý průmět podstavy do 2D roviny).

`volume_f_rbh_raster.py` – druhý přístup k výpočtu objemu. Vstupem jsou footprinty budov a rastr relativních výšek budov. Pro každou buňku rastru je vytvořen bod, spojený unikátním identifikátorem s footprintem budovy. Bod převezme Z souřadnici rastru a ze všech bodů unikátní budovy je vypočten průměr Z hodnoty, tedy průměrná výška budovy. Ta je následně vynásobena plochou footprintu budovy, čímž vzniká přibližná

aproximace objemu, která je ovšem méně přesná než výpočet skrze 3D model budovy. Výpočet tímto způsobem trvá řádově několikrát méně času než výpočet z 3D modelu.

5.2 Výpočet objemu skrze PostGIS

Tento postup je open source alternativa metody použité ve skriptu `volume_f_rbh_raster.py`. Skripty, psané v Pythonu 3.6, jsou rovněž dostupné na zmiňovaném GitHubu. Výhodou je vyšší rychlost výpočtu.

`database_calc.py` – vstupními parametry je databázové schéma, tabulka s footprinty budov, sloupeček s jejich geometrií, tabulka s rastrem relativních výšek budov (RUM defaultně při importu převádí rastr na polygony) a výstupní tabulka. V první řadě se inicializuje připojení k databázi skrze konfigurační soubor `dbconn.JSON`. Nejprve se ke každému footprintu vytvoří unikátní ID. Následně, obdobně jako v předchozím skriptu, je ke každému pixelu rastru vygenerován bod, obsahující informaci o výšce. Z těchto bodů, spojených s budovou unikátním ID budovy, je vypočten průměr, který po vynásobení s plochou footprintu udá přibližnou aproximaci objemu. Nyní má každá budova atribut svého objemu a v další části se vytvoří centroid budovy. Ten je na základě své geometrie přiřazen ke čtverci mřížky. Výpočet tímto způsobem trval přibližně pět hodin.

```
1. import psycopg2
2. import argparse
3. import json
4. import datetime
5.
6. parser = argparse.ArgumentParser(description = "This script
   calculates volume from footprints of buildings and raster with
   relative heights of the buildings. Should be used as extension of
   the ReconfigurableUrbanModeler (https://github.com/simberaj/rum).
   All inputs should share the same SRID as the grid of the RUM.")
7. parser.add_argument('-s', '--database_schema', type = str, metavar
   = '', required = True, help = 'Database schema.')
8. parser.add_argument('-d', '--input_footprints', type = str, metavar
   = '', required = True, help = 'Name of the table with bulding
   footprints.')
9. parser.add_argument('-c', '--area_column', type = str, metavar =
   '', required = True, help = 'Name of the column with area in the
   input footprint table.')
10.  parser.add_argument('-r', '--input_raster', type = str, metavar =
   '', required = True, help = 'Name of the table with raster with
   relative heights of the buildings.')
11.  parser.add_argument('-o', '--output_table', type = str, metavar =
   '', required = True, help = 'Name of the output table')
12.  args = parser.parse_args()
13.  schema = str(args.database_schema)
14.
15.  with open('dbconn.JSON') as config_file:
16.      login = json.load(config_file)
```

```
17.
18.  now = datetime.datetime.today()
19.  print(str(now) + " Starting the script, connecting to database.")
20.
21.  class DatabaseTask:
22.      schemaSQL = str(args.database_schema)
23.      foot = str(args.input_footprints)
24.      rastr = str(args.input_raster)
25.      output = str(args.output_table)
26.      area_building = str(args.area_column)
27.      def __init__(self):
28.          try:
29.              self.connection = psycopg2.connect(**login)
30.              self.connection.autocommit = True
31.              self.cursor = self.connection.cursor()
32.              now = datetime.datetime.today()
33.              print(str(now) + " Connected to database.")
34.          except:
35.              now = datetime.datetime.today()
36.              print(str(now) + " Cannot connect to the database.")
37.
38.      def add_id(self):
39.          schema = self.schemaSQL
40.          buildings = self.foot
41.          self.cursor.execute(f'ALTER TABLE {schema}.{buildings}
   ADD COLUMN uniqueid int GENERATED BY DEFAULT AS IDENTITY')
42.
43.      def calc_points(self):
44.          schema = self.schemaSQL
45.          rastr = self.rastr
46.          buildings = self.foot
47.          self.cursor.execute(f'CREATE TABLE {schema}.bodyras (id
   integer, height float,geom geometry)')
48.          self.cursor.execute(f'INSERT INTO
   {schema}.bodyras(height, geom) SELECT band_1,
   ST_GeneratePoints(geometry, 100) FROM {schema}.{rastr}')
49.          self.cursor.execute(f'UPDATE {schema}.bodyras SET id =
   "uniqueid" FROM {schema}.{buildings} WHERE
   ST_Contains({buildings}.geometry, bodyras.geom)')
50.          now = datetime.datetime.today()
51.          print(str(now) + " Points calculated.")
52.
53.      def calc_volume(self):
54.          schema = self.schemaSQL
55.          output = self.output
56.          buildings = self.foot
57.          area = self.area_building
58.          self.cursor.execute(f'CREATE TABLE {schema}.{output} AS
   (SELECT id, AVG(bodyras.height) FROM {schema}.bodyras WHERE id IS
   NOT NULL GROUP BY id)')
59.          self.cursor.execute(f'DROP TABLE {schema}.bodyras')
60.          self.cursor.execute(f'ALTER TABLE {schema}.{output} ADD
   COLUMN volumev float')
61.          self.cursor.execute(f'ALTER TABLE {schema}.{output} ADD
   COLUMN geom geometry')
62.          self.cursor.execute(f'ALTER TABLE {schema}.{output} ADD
   COLUMN areafoot float')
63.          self.cursor.execute(f'UPDATE {schema}.{output} SET geom =
   {buildings}.geometry FROM {schema}.{buildings} WHERE id =
   "uniqueid"')
```

```
64.          self.cursor.execute(f'UPDATE {schema}.{output} SET
   areafoot = {area} FROM {schema}.{buildings} WHERE id = "uniqueid"')
65.          self.cursor.execute(f'UPDATE {schema}.{output} SET
   volumev = ("avg"*areafoot)')
66.          now = datetime.datetime.today()
67.          print(str(now) + " Volume calculated.")
68.
69.      def centroid_prepare(self):
70.          schema = self.schemaSQL
71.          output = self.output
72.          self.cursor.execute(f'CREATE TABLE {schema}.table1 AS
   ((SELECT geohash FROM {schema}.grid))')
73.          self.cursor.execute(f'ALTER TABLE {schema}.table1 ADD
   COLUMN volume float')
74.          self.cursor.execute(f'ALTER TABLE {schema}.table1 ADD
   COLUMN geom geometry')
75.          self.cursor.execute(f'UPDATE {schema}.table1 SET geom =
   grid.geometry FROM {schema}.grid WHERE grid.geohash =
   table1.geohash')
76.          self.cursor.execute(f'UPDATE {schema}.table1 SET volume =
   volumev FROM {schema}.{output} WHERE ST_Contains(table1.geom,
   (ST_Centroid({output}.geom)))')
77.          now = datetime.datetime.today()
78.          print(str(now) + " Centroids prepared.")
79.
80.
81.      def centroid_feat(self):
82.          schema = self.schemaSQL
83.          output = self.output
84.          self.cursor.execute(f'CREATE TABLE {schema}.table2 AS
   (SELECT id, volumev, ST_Centroid({output}.geom) FROM
   {schema}.{output})')
85.          self.cursor.execute(f'ALTER TABLE {schema}.table2 ADD
   COLUMN geohash text;')
86.          self.cursor.execute(f'UPDATE {schema}.table2 SET geohash
   = grid.geohash FROM {schema}.grid WHERE ST_contains(grid.geometry,
   table2.st_centroid)')
87.          self.cursor.execute(f'CREATE TABLE
   {schema}.feat_building_3 AS (SELECT geohash, SUM(volumev) AS
   volume_PostGIS FROM {schema}.table2 GROUP BY geohash)')
88.          self.cursor.execute(f'DROP TABLE {schema}.table1;')
89.          self.cursor.execute(f'DROP TABLE {schema}.table2;')
90.          now = datetime.datetime.today()
91.          print(str(now) + " Feat table prepared.")
92.
93.
94.  if __name__ == '__main__':
95.      database_operation = DatabaseTask()
96.      database_operation.add_id()
97.      database_operation.calc_points()
98.      database_operation.calc_volume()
99.      database_operation.centroid_prepare()
100.     database_operation.centroid_feat()
101.     now = datetime.datetime.today()
102.     print(str(now) + " Script completed, volume calculated.")
```

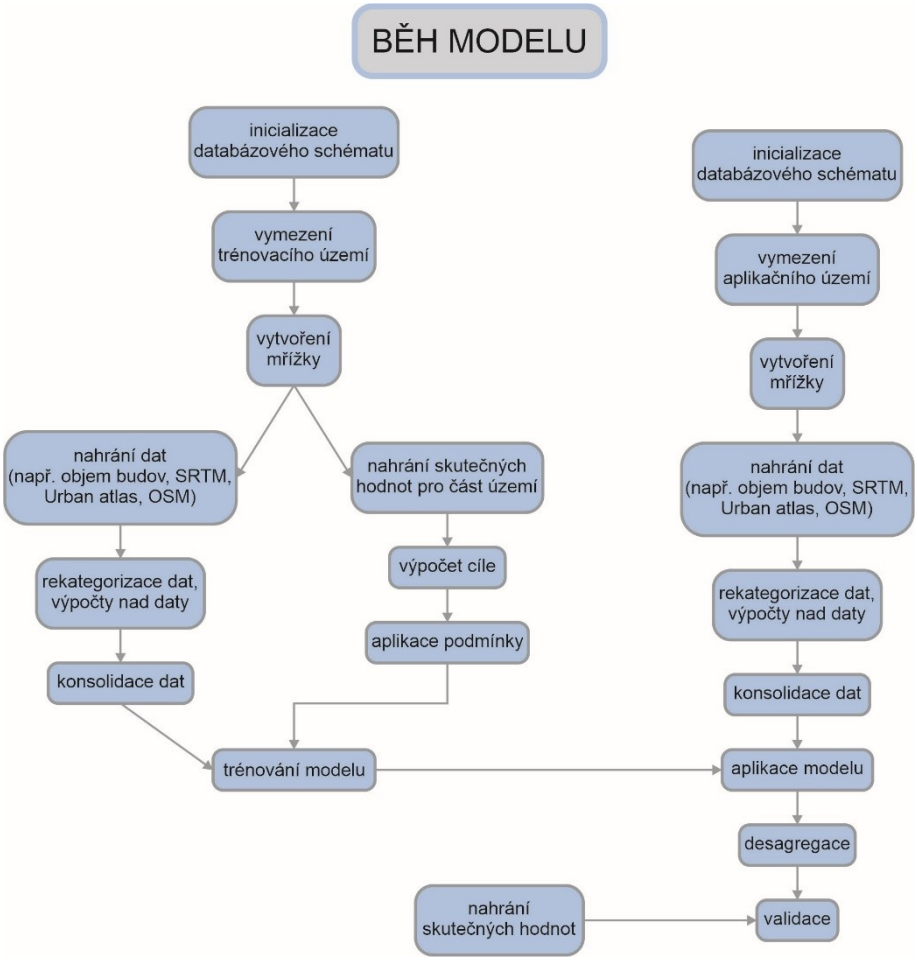
Z toho vyplývá, že pokud budova spadá do dvou či tří čtverců mřížky, ve výsledku je její centorid zařazen pouze do jednoho čtverce. Tento problém by mohl být vyřešen "rozřezáním" budovy dle čtverců mřížky a výpočtem objemu pro každou část budovy.

`create_feat.py` – vytváří `feat_` tabulku, která je nutná pro zahrnutí do trénovacích parametrů modelu, ze vstupních dat, která již mají vypočtený objem.

5.3 Spuštění modelu

Praktické modelování probíhalo dle workflow desagregace (viz obrázek č. 2).

Obrázek č. 2: Schéma chodu modelu. Zdroj: Šimbera 2020, vlastní tvorba



Do trénování i aplikace vstupuje tabulka `all_feats`, vzniklá konsolidací všech zvolených dostupných dat do jedné tabulky. Výběrem dat, které jsou do tabulky zařazena, upravujeme konfiguraci modelu. Na samotné trénování modelu byla využita učící metoda Random forest, jež vytváří více rozhodovacích stromů a následně vydává modus tříd, vrácených jednotlivými stromy. Random forest byl využit a doporučen i Šimberou (2020), neboť v porovnání s dalšími metodami (lineární regrese, Lasso etc.) podává nejlepší výsledky. Důležitost jednotlivých parametrů, tedy sloupečků v tabulce `all_feats`, lze

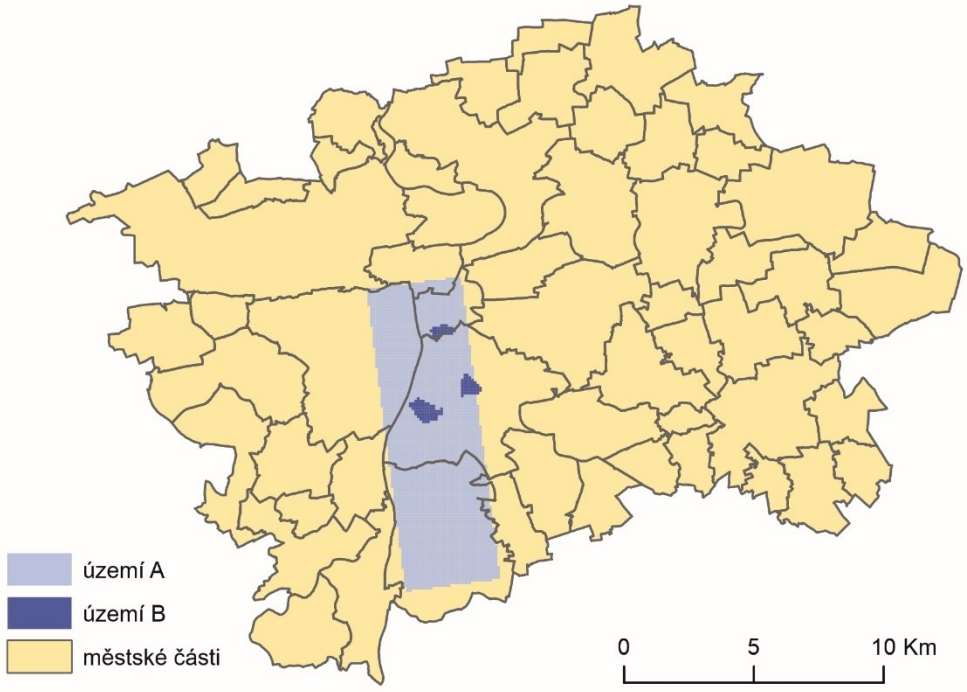
z modelu vypsat skriptem `introspect_model.py`. Výsledkem jsou relativní hodnoty v intervalu [0 %; 100 %], udávající důležitost parametrů tak, jak byly vypočteny v rámci trénovacích dat.

5.3.1 Trénovací území

Výběr správného trénovacího území je klíčový. Pokud bychom model aplikovali na jiné město než na Prahu, trénovacím územím by byla Praha. Zde ovšem model trénujeme i aplikujeme na Prahu, proto musela být vybrána část Prahy, na které byl model natrénován.

Trénovací území by mělo zahrnovat všechny typy zástavby, se kterými se model může při desagregaci setkat. Pokud by některý typ krajiny či zástavby chyběl, mohlo by dojít k situaci, že model nebude mít informaci, jak se v určité kompozici zachovat a odhad bude nesprávný.

Obrázek č. 3: Trénovací území. Zdroj: IPR (2019c), vlastní tvorba



Nejprve bylo vybráno území A, tedy pás táhnoucí se od Modřan přes Braník až na Nové Město. Zahrnuje i části Radlic a Smíchova na druhé straně Vltavy. Území A zahrnuje výškově členitou krajinu, od které se odvíjí rozmanité druhy zástavby.

Aby byla otestována robustnost modelu, tedy zdali význam informace obsažené v objemu budov roste se zmenšujícím se trénovacím územím, byly vybrány tři základní sídelní jednotky, tvořící území B. Nuselské údolí, jako zástupce typické městské zástavby z počátku

20. století, sídliště Na Zelené lišce a Braník – Na Křížku jako čtvrť s vilovými domky. Na území B chybí některé podstatné druhy zástavby jako velké sklady, chatařské oblasti, parky, velké silnice, železnice, hřbitovy či lesy, což povede k horším výsledkům modelů trénovaných na tomto území. Na druhou stranu lze takto popsat chování modelu v případě dostupnosti trénovacích dat pro velmi malé území.

5.4 Validace

Pro validaci modelu je nutné znát skutečné hodnoty a dopočítat jejich hodnoty pro čtverce mřížky. Následně je porovnán výstup z modelu, tedy odhadnutá hodnota a hodnota skutečná. Skript `validate.py` nabízí několik statistik, popisujících přesnost modelu.

- **DIFF** – Ukazuje absolutní rozdíl mezi počtem obyvatel, jenž je desagregována, a mezi součtem všech skutečných hodnot.
- **TAE (total absolute error)** – Udává absolutní hodnotu počtu obyvatel, kteří byli modelem odhadnuti správně. m_i představuje reálné hodnoty počtu obyvatel pro každý čtverec.

$$\mathrm{TAE} = \sum \mathrm{m_i} * \mathrm{RTAE}$$

- **RTAE (relative total absolute error) –** Měří celkovou velikost chyby, normalizovanou sumou skutečného počtu obyvatel.

$$\mathrm{RTAE} = \frac{1}{\sum_i \mathrm{m_i}} \sum_i |\hat{\mathrm{m}}_i - \mathrm{m_i}|$$

Hodnota m_i opět představuje reálné počty obyvatel pro každý čtverec, zatímco $\hat{m}_i$ hodnoty odhadnuté, tedy vypočtené modelem. RTAE nabývá hodnot v intervalu [0 %; 200 %], kde 0 % znamená naprostou shodu a dokonalou přesnost modelu. Naproti tomu v případě 200 % by byla všechna populace přiřazena do čtverců s nulovou skutečnou populací. RTAE je využívána více autory, například Šimberou (2020), Rosinou et al. (2017) či Wangem et al. (2016). Bakilah et al. (2014) využívá obdobnou metriku označovanou jako mean absolute error (MAE).

- **R2 (koeficient determinace) –** Hodnotí míru kvality modelu. Nabývá hodnot v intervalu [0; 100], kde 100 značí nejlepší shodu.

$$R^2 = 1 - \frac{\sum_i (\hat{m}_i - m_i)^2}{\sum_i (m_i - \bar{m}_i)^2}$$

- **RMSE (root mean squared error) –** Udává míru kvality modelu. Čím je výsledná hodnota blíže 0, tím je model přesnější.

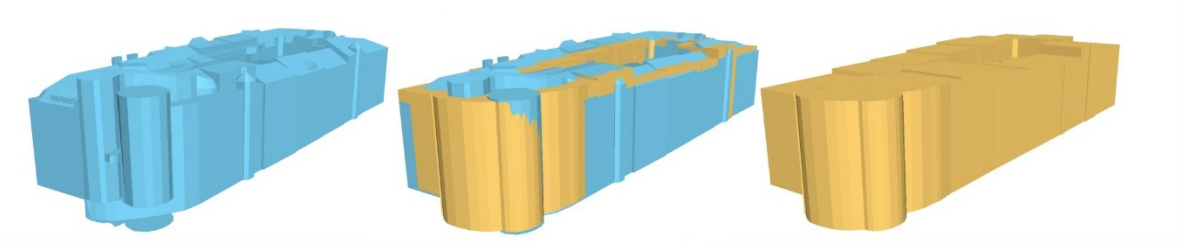
$$\mathrm{RMSE} = \sqrt{\frac{\sum_i (\hat{m}_i - m_i)^2}{\sum_i m_i}}$$

6 Výsledky a diskuse

6.1 Výpočet objemu

První cíl práce, výpočet objemu z 3D modelu města Prahy, byl realizován třemi způsoby. Každý způsob přinesl jiný výsledek s ohledem na použitou metodu. Vzhledem k formátům dat, ve kterých jsou 3D budovy města Prahy (IPR 2019a) distribuovány, a to sice DWG, DNG a Shapefile, byla vrstva 3D budov využita pouze pro jeden způsob výpočtu. Sérií skriptů, popsaných v metodické části práce, byla data ve formátu Shapefile multipatch uzavřena, zkontrolována jejich uzavřenost a vypočten objem. Poté byly z 3D budov vytvořeny footprinty s informací o objemu, která byla zahrnuta do modelování. Touto metodou bylo vypočteno, že objem všech budov v Praze činí 608 663 669 m^3, tedy přibližně jako dvě vodní nádrže Lipno. Průměrný objem jedné budovy je 2 666 m^3, což odpovídá krychli se stranou o délce přibližně 13,86 m.

Obrázek č. 4: Tančící dům. Modře multipatch, oranžově model vytvořený z rastru relativních výšek budov. Zdroj: IPR (2019a), IPR (2017), vlastní tvorba

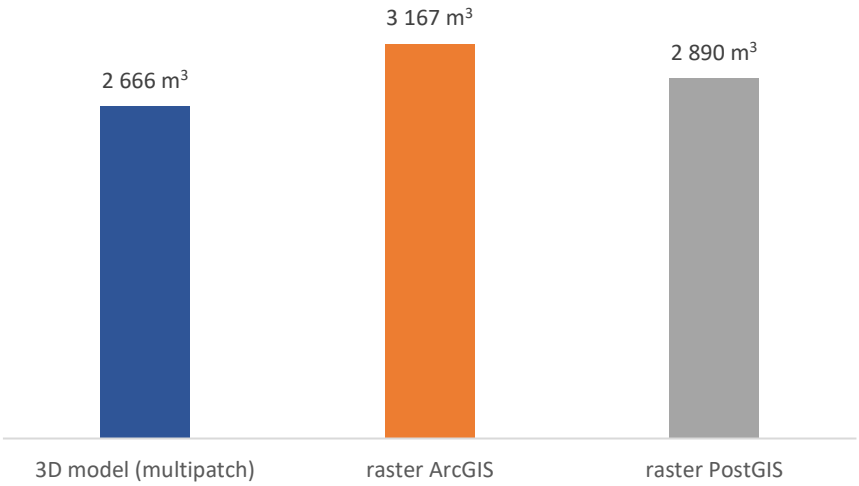


Druhou metodou výpočtu se stal výpočet z rastru relativních výšek budov. Byly využity footprinty budov z digitální technické mapy Prahy (IPR 2020). Touto metodou vychází průměrný objem budovy na 3 167 m^3. Rozdíl můžeme vysvětlit tím, že tato metoda zanedbává veškeré římsy, převisy a dutiny, nad nimiž je další patro, či střecha. Rovněž je zanedbán přesah střechy. V případě některých staveb to může uměle zvýšit vypočtenou hodnotu objemu (viz Tančící dům na obrázku č. 4). Velká část staveb je velmi malých (chaty,

garáže, objekty ve vnitroblocích), proto průměrná plocha stavby v Praze činí 187 m^2. Vzhledem k velikosti pixelu rastru jeden metr můžeme očekávat větší chybovost a nadhodnocování zejména u malých staveb, jako zmíněné objekty či rodinné domy, vily etc.

Stejná metodika výpočtu mohla být realizována i skrze PostGIS sérii SQL příkazů, inicializovaných ve skriptu `database_calc.py`. Výhoda tohoto třetího způsobu výpočtu objemu oproti předchozím dvěma je v jeho aplikovatelnosti, kde využívá pouze Python 3.6 a PostgreSQL databázi s extenzí PostGIS. Aplikační rámec tedy koresponduje s RUMem a je kompletně Open source. Po aplikaci na budovy OSM byl vypočten průměrný objem budovy 2 890 m^3 (viz obrázek č. 5), tedy podobně jako v případě multipatche.

Obrázek č. 5: Průměrné objemy budov v Praze s použitím tří různých metod. Zdroj: GEOFABRIK (2020), IPR (2020), IPR (2019a), IPR (2017), vlastní tvorba



Využití 3D modelu pro výpočet objemu můžeme považovat za nejpřesnější, nicméně udávaná přesnost jeden metr a úprava modelu nástrojem Enclose multipatch vypočtenou hodnotu rovněž zkresluje. Z hlediska rychlosti výpočtu je nejrychlejší PostGIS, kde ovšem stále mluvíme o několika hodinách. Rozdíl mezi výpočtem z rastru v ArcGISu a PostGISu je takový, že v ArcGISu lze nastavit maximální rozestup mezi body, jenž se generují a přejímají informaci o výšce z rastru. V PostGISu lze definovat pouze počet bodů, jenž bude vygenerován pro každou budovu. Proto trvá výpočet pro každou budovu stejně dlouhý čas, zatímco v ArcGISu je čas výpočtu závislý na počtu pixelů rasteru, jenž překrývá budovu.

3D model města Prahy ve formě, ve které je distribuován, lze využít k vizualizaci, nicméně analytické využití vyžaduje uzavřenost modelu. Při tvorbě modelu by měl být kladen důraz nejen na jeho vizuální stránku, ale i na funkčnost.

Biljecki et al. (2016) využívá ke generování 3D modelu mračno bodů, zmapované pro celé území Nizozemska. Celkem se jedná o 639 miliard bodů, ze kterých jsou nad footprinty budov metodou extruze vytvořeny jejich 3D modely. Podobné mračno bodů, dostupné pro celé Česko, by mohlo posloužit k případné desagregaci nejen na úrovni Prahy, ale v celorepublikovém měřítku.

6.2 Aplikace modelu

Cílem bylo zjistit, jestli využití 3D modelu při desagregaci přinese zpřesnění výsledků, případně jak výrazné toto zlepšení bude. Abychom mohli porovnat výsledky, musela být nejprve provedena desagregace nad 2D daty, tedy prakticky zopakován postup Jana Šimbery (2020). Výsledky uvedené práce nelze ovšem srovnat s přímo s výsledky aplikace modelu s 3D daty, neboť Jan Šimbera model aplikuje na město Maribor. Velikost mřížky, do které jsou hodnoty desagregovány, byla ponechána na defaultních 100x100 metrů. Celkem bylo provedeno více než 70 aplikací modelu s různými parametry, různou velikostí mřížky a na různém trénovacím území.

6.2.1 Aplikace nad 2D daty

Z důvodu dosažení objektivní porovnatelnosti byla desagregace provedena nad 2D daty. Tato data, popsaná v metodické části, jsou rámcově stejná jako data využitá Janem Šímberou (2020). Liší se ovšem časový horizont dat. Celkem tato data tvoří 41 sloupečků v tabulce `all_feats`, tedy 41 parametrů. Rozdělena jsou do pěti skupin dle zdroje parametru.

- ***urbanatlas*** – zahrnuje rekategorizovaný Urban Atlas (Evropská komise 2012).
- ***building*** – budovy z OSM (Geofabrik 2020) a indexy z těchto budov vypočítané.
- ***poi*** – body zájmu z OSM (Geofabrik 2020).
- ***transport*** – komunikace z OSM (Geofabrik 2020) a indexy z těchto komunikací vypočítané.
- ***srtm*** – sklon svahu a orientace vůči světovým stranám, vypočítané z digitálního modelu terénu (USGS 2020).

Při trénování nad územím A (viz tabulka č. 1) model natrénoval následující důležitost parametrů a jejich skupin.

Tabulka č. 1: Důležitost prvních deseti nejdůležitějších parametrů a jejich skupin v rámci modelu s 2D daty (území A). Zdroj: vlastní tvorba

Parametr	Důležitost
Urban Atlas třída 111	30,93 %
Budovy - průměrná plocha	15,49 %
Budovy - plocha	6,79 %
POI - ubytování	6,61 %
Budovy - index konkávnosti	4,63 %
Délka pěších cest	3,25 %
POI - Práce	3,07 %
SRTM - Sklon svahu	2,47 %
Délka obslužných komunikací	2,34 %
POI - Vybavenost	2,31 %

Dle zdroje parametru	Důležitost
Urban Atlas	36,56 %
Budovy	33,34 %
Body zájmu	15, 39 %
Komunikace	10, 46 %
Digitální mode terénu	4,25 %

Dle tabulky č. 1 má nejvyšší důležitost pro rozložení cílového zkoumaného jevu, tedy počtu obyvatel v tomto případě Urban Atlas třída 111. Tato třída je definována jako souvislá městská zástavba, kde více jak 80 % povrchu tvoří budovy (Evropská komise 2016). Druhým nejdůležitějším parametrem je průměrná plocha budovy na jeden čtverec mřížky a třetím součet ploch budov na čtverec mřížky. Zde je patrné, že tzv. feature engineering, kdy je z určité hodnoty dat pro mřížku vypočtena hodnota relativní, může být pro odhad sledovaného jevu užitečnější než hodnota původní, neboť průměrná plocha budovy je v tomto případě důležitějším ukazatelem pro rozložení populace než celková plocha zástavby ve čtverci.

Tabulka č. 2: Výsledky desagregace nad 2D daty. Zdroj: vlastní tvorba

Parametry	Území	Mřížka	RTAE	R2	RMSE
2D data (41 parametrů)	A	100 m	58,37 %	64,58 %	40,65 %
2D data (41 parametrů)	B	100 m	106,65 %	25,18 %	59,09 %
2D data bez Urban Atlasu (32 parametrů)	A	100 m	83,37 %	44,95 %	50,68 %
2D data bez Urban Atlasu (32 parametrů)	B	100 m	108,30 %	23,41 %	59,78 %

Urban Atlas není dostupný pro celé území Česka, proto byl model proveden i v konfiguraci, kdy při modelování Urban Atlas zahrnut nebyl. Cílem bylo zjistit, jak se bude lišit odhad populace bez tohoto datového zdroje. Pokud by byla prováděna celostátní studie, podobně jak činil Biljecki et al. (2016), nebyl by Urban Atlas pro většinu území dostupný.

Hodnota RTAE (viz tabulka č. 2) vyšla při trénování na větším území A 58,37 %. Bez Urban Atlasu je tato hodnota ovšem mnohem horší, a to 83,37 %. Při trénování na menším území B vychází RTAE 106,65 % bez Urban Atlasu s 108,3 % s Urban Atlasem. Při zmenšení trénovacího území tedy klesá důležitost Urban Atlasu v modelování, nicméně se vzrůstající plochou trénovacího území roste.

6.2.2 Aplikace nad 3D daty

Informace o objemu budov, získaná z 3D dat, je koncipována jako doplněk k 2D datům. Aby mohl být objem, jenž byl vypočten pro každou budovu, začleněn do parametrů modelování, musela být hodnota agregována do čtverců mřížky. Velká část budov se nachází ve dvou či ve třech čtvercích mřížky (viz obrázek č. 6), proto muselo být rozhodnuto, k jakému čtverci bude objem budovy připočten.

Pokud se budova nachází ve více čtvercích, měl by být i její 3D model, a tedy i vypočtený objem rozdělen do těchto čtverců na základě průsečíku mřížky a 3D modelu budovy. Rozdělení footprintů může být realizováno nástrojem Split. Tento nástroj ovšem z footprintů náležících do jednoho čtverce vytvoří jeden shapefile, počet čtverců mřížky je tedy roven počtu vytvořených shapefilů. V případě mřížky o velikosti čtverce 100x100 m se jedná o 50 547 shapefilů, jejichž vytvoření zabralo přibližně 24 hodin. Dalším krokem by bylo sjednocení shapefilů, například nástrojem Merge, a jejich použití pro výpočet objemu z rastru relativních výšek budov. Práce s takovým množstvím souborů je nicméně hardwarově enormně náročná a další zpracování se ukázalo být na osobním počítači časově nemožné. PostGIS nabízí podobný nástroj ST_Split.

Ve 3D je situace ještě složitější než s footprinty, neboť v rámci toolboxu 3D Analyst neexistuje nástroj Split, nicméně stále by mohlo být dělení 3D modelu realizováno, například nástrojem Intersect 3D.

Obrázek č. 6: Zástavba s centroidy nad mřížkou. Zdroj: IPR (2019a), vlastní tvorba

Z výše uvedených důvodů proběhla realizace generováním centroidů, kde každý centroid náleží právě do jednoho čtverce. Vzhledem k početní náročnosti bylo v rámci této

práce od rozdělování 3D modelu dle mřížky upuštěno, nicméně skýtá se zde možnost vylepšení výsledku modelu v případě informace za části budov, a ne za jejich celky, převedené v centroidy. Zanesená chyba je částečně vyvažována, neboť pokud je část budovy zařazena špatně do vedlejšího čtverce, s určitou pravděpodobností bude opět část budovy z vedlejšího čtverce zařazena chybně.

Celkem byly vypočteny čtyři charakteristiky objemu, jenž byly nadále v modelování kombinovány, aby bylo dosaženo co nejlepšího výsledku.

- 3D model (multipatch) – objem vypočtený z 3D modelu
- Raster ArcGIS – objem vypočtený z rastru relativních výšek budov skrze ArcGIS
- Raster PostGIS – objem vypočtený z rastru relativních výšek budov skrze PostGIS
- AVG 3D model – objem vypočtený z 3D modelu, kde není čtverci mřížky přiřazena suma objemu budov, ale průměr objemu pro daný čtverec

Důležitosti parametrů, jež model natrénoval na území B jsou zobrazeny v tabulce č. 3. V tabulce č. 4 jsou zachyceny nejdůležitější parametry po trénování na území A.

Tabulka č. 3: Důležitost prvních pěti nejdůležitějších parametrů v rámci modelu s 3D modelem (multipatch) a s objemem, spočítaným z rastru relativních výšek budov v ArcGISu. (území B).
Zdroj: vlastní tvorba

3D model (multipatch)	B
Parametr	Důležitost
3D model (multipatch)	77,20 %
Sklon svahu	3,66 %
Délka obslužných komunikací	3,02 %
Urban Atlas třída 111	1,97 %
Budovy - plocha	1,79 %

Raster ArcGIS	B
Parametr	Důležitost
Raster ArcGIS	74,07 %
Urban Atlas třída 111	3,64 %
Budovy - plocha	3,57 %
Délka obslužných komunikací	2,62 %
Délka ulic	2,57 %

Tabulka č. 4: Důležitost prvních deseti nejdůležitějších parametrů v rámci modelu s 3D daty (území A). Zdroj: vlastní tvorba

3D model (multipatch)	A
Parametr	Důležitost
3D model (multipatch)	39,59 %
Urban Atlas třída 111	9,97 %
Urban Atlas třída 121	7,41 %
POI - ubytování	7,19 %
Budovy - plocha	4,63 %
Budovy - průměrná plocha	3,25 %
Délka pěších cest	3,07 %
Budovy - index konkávnosti	2,47 %
Délka obslužných komunikací	2,34 %
POI - Práce	2,31 %

Raster ArcGIS	A
Parametr	Důležitost
Raster ArcGIS	38,89 %
Urban Atlas třída 111	12,40 %
Urban Atlas třída 121	6,33 %
POI - ubytování	5,94 %
Budovy - plocha	4,07 %
Budovy - průměrná plocha	3,61 %
Budovy - index konkávnosti	2,60 %
POI - Práce	2,57 %
Délka pěších cest	2,57 %
Sklon svahu	2,19 %

AVG 3D model	A
Parametr	Důležitost
Urban Atlas třída 111	28,49 %
AVG 3D model	23,08 %
POI - ubytování	8,93 %
Budovy - plocha	4,36 %
POI - Práce	3,06 %
Budovy - index konkávnosti	2,46 %
Sklon svahu	2,40 %
Délka obslužných komunikací	2,38 %
Budovy - průměrná plocha	2,30 %
Urban Atlas třída 112	2,23 %

Raster PostGIS	A
Parametr	Důležitost
Urban Atlas třída 111	27,55 %
Budovy - průměrná plocha	16,86 %
Budovy - plocha	5,60 %
POI - ubytování	5,54 %
Raster PostGIS	4,66 %
Budovy - index konkávnosti	3,51 %
Délka pěších cest	3,03 %
POI - Práce	2,98 %
Délka obslužných komunikací	2,82 %
POI - vybavenost	2,65 %

V případě území A je při trénování s objemem vypočteným z multipatche a s objemem vypočteným v ArcGISu z rastru relativních výšek budov považován tento parametr modelem za nejdůležitější pro odhad sledovaných hodnot. Tvoří téměř 40 % ze všech 42 parametrů. V případě modelu s objemem, vypočteným v PostGISu má ovšem objem pouze necelých 5 % důležitosti v modelování. Stejně jako v případě rozdílu průměrného objemu, vypočteného z rastru v ArcGISu a v PostGISu rozdíl patrně vzniká použitím různých footprintů budov a rozdílnou metodou generování bodů, z nichž je výška budovy odečítána. Dalším klíčovým faktorem, ovlivňujícím přesnost metody počítané skrze PostGIS je počet bodů, jež byl generován. Ten byl kvůli omezenému množství dostupné paměti a výkonu nastaven pouze na 5 bodů pro každou budovu. Se zvyšujícím počtem bodů by rostla přesnost, a tedy i důležitost výpočtu objemu. Feature engineering, výpočet průměrného objemu na čtverec mřížky z 3D modelu získal 23 % důležitosti, tedy druhou nejvyšší důležitost za Urban Atlas třídou 111. Tato třída se stejně jako u 2D dat osvědčila jako hodnotný ukazatel rozložení sledovaného jevu.

Při trénování na území B ovšem důležitost objemu roste, objem vypočtený z 3D modelu zde má důležitost 77,2 % a objem vypočtený v ArcGISu z rastru relativních výšek budov 74,07 %. Trénováním na menším území tedy roste důležitost objemu.

Tabulka č. 5: Výsledky desagregace nad 3D daty. Zdroj: vlastní tvorba

Parametry	Území	Mřížka	RTAE	R2	RMSE
2D data + 3D model (42 parametrů)	A	100 m	57,59 %	67,90 %	38,70 %
2D data + Raster ArcGIS (42 parametrů)	A	100 m	58,01 %	67,15 %	39,15 %
2D data + Raster PostGIS (42 parametrů)	A	100 m	59,83 %	66,10 %	39,78 %
2D data + AVG 3D model (42 parametrů)	A	100 m	62,59 %	64,58 %	40,66 %
2D data bez Urban Atlasu + 3D model (33 parametrů)	A	100 m	66,47 %	60,33 %	43,02 %
2D data bez OSM + 3D model (12 parametrů	A	100 m	63,40 %	64,04 %	40,96 %
2D data + 3D model (42 parametrů)	B	100 m	94,95 %	37,52 %	54,00 %
2D data + Raster ArcGIS (42 parametrů)	B	100 m	97,84 %	35,44 %	54,89 %
2D data + Raster PostGIS (42 parametrů)	B	100 m	98,68 %	34,57 %	55,26 %
2D data + AVG 3D model (42 parametrů)	B	100 m	96,28 %	34,73 %	55,19 %
2D data bez Urban Atlasu + 3D model (33 parametrů)	B	100 m	97,90 %	33,41 %	55,75 %
2D data bez OSM + 3D model (12 parametrů)	B	100 m	91,83 %	39,51 %	53,13 %

Nejlepších výsledků model trénovaný na území A dosáhl při zahrnutí objemu vypočítaného z 3D modelu. RTAE vyšlo 57,59 %, tedy přibližně o procento lépe než bez 3D modelu (viz tabulka č. 5). Mírné zlepšení nastalo i v případě ArcGIS rastru. Celkové zlepšení o takto malé hodnoty je zřejmě zapříčiněno tím, že v třídách Urban Atlasu již určitý rozdíl ve výšce budov a hustotě zástavby obsažen je (například mezi třídou 111 a 112). Model s vypočteným objemem skrze PostGIS dosáhl v případě trénovacího území A podobných výsledků jako model s 2D daty. Výpočet průměru z 3D modelu se zde neosvědčil, výsledek je dokonce horší než model natrénovaný pouze s 2D daty.

Zajímavějších výsledků dosáhl model trénovaný nad územím A při nezahrnutí Urban Atlasu. Zatímco 2D data bez Urban Atlasu dosáhla RTAE 83,37 %, pokud byla k těmto datům přidána informace o objemu z 3D modelu, RTAE kleslo na 66,47 %. Pokud naopak do trénování byl zahrnut Urban Atlas, ale nebyla zahrnuta data z OSM, RTAE vyšlo 63,40 %. Informaci o objemu tedy můžeme považovat za hodnotnější, pokud chybí jiný datový zdroj, například Urban Atlas.

Při trénování modelu nad podstatně menším územím B vyšel rozdíl mezi 2D daty a daty s objemem podstatně větší. Zatímco 2D data získala 106,65 %, při zahrnutí objemu RTAE vyšlo 94,95 %. Hodnota pod 100 % vyšla pro všechny kombinace objemů, ale i pro kombinaci bez Urban Atlasu. Ta získala RTAE 97,90 %, na tomto trénovacím území tedy

informace o objemu nahradila data Urban Atlasu, a dokonce dosáhla lepších výsledků. Nejlepší výsledek vyšel pro model, trénovaný s objemem z 3D modelu, ale bez OSM. Zde bylo RTAE 91,83 %. Informace o budově, obsažené v OSM, zde byly nahrazeny 3D modelem a další prvky, jako POI či komunikace, pravděpodobně nemají u malého trénovacího území požadovaný efekt.

6.2.3 Změna velikosti mřížky

Dosavadní modelování probíhalo vždy pro mřížku o délce strany čtverce 100 m. Další zkoumanou skutečností byla závislost výstupů modelu na změně velikosti mřížky. Při aplikaci nad 200 m mřížkou (viz tabulka č. 6) se RTAE s 2D daty zlepší o přibližně 4 %, avšak s 3D daty přibližně o 12 %. O několik procent přesnějších výsledků je dosaženo po aplikaci na 500 m mřížku, nicméně aplikací na 1 000 m mřížku se výsledky zhoršují. Dáno je to zřejmě výškovou a architektonickou rozmanitostí Prahy.

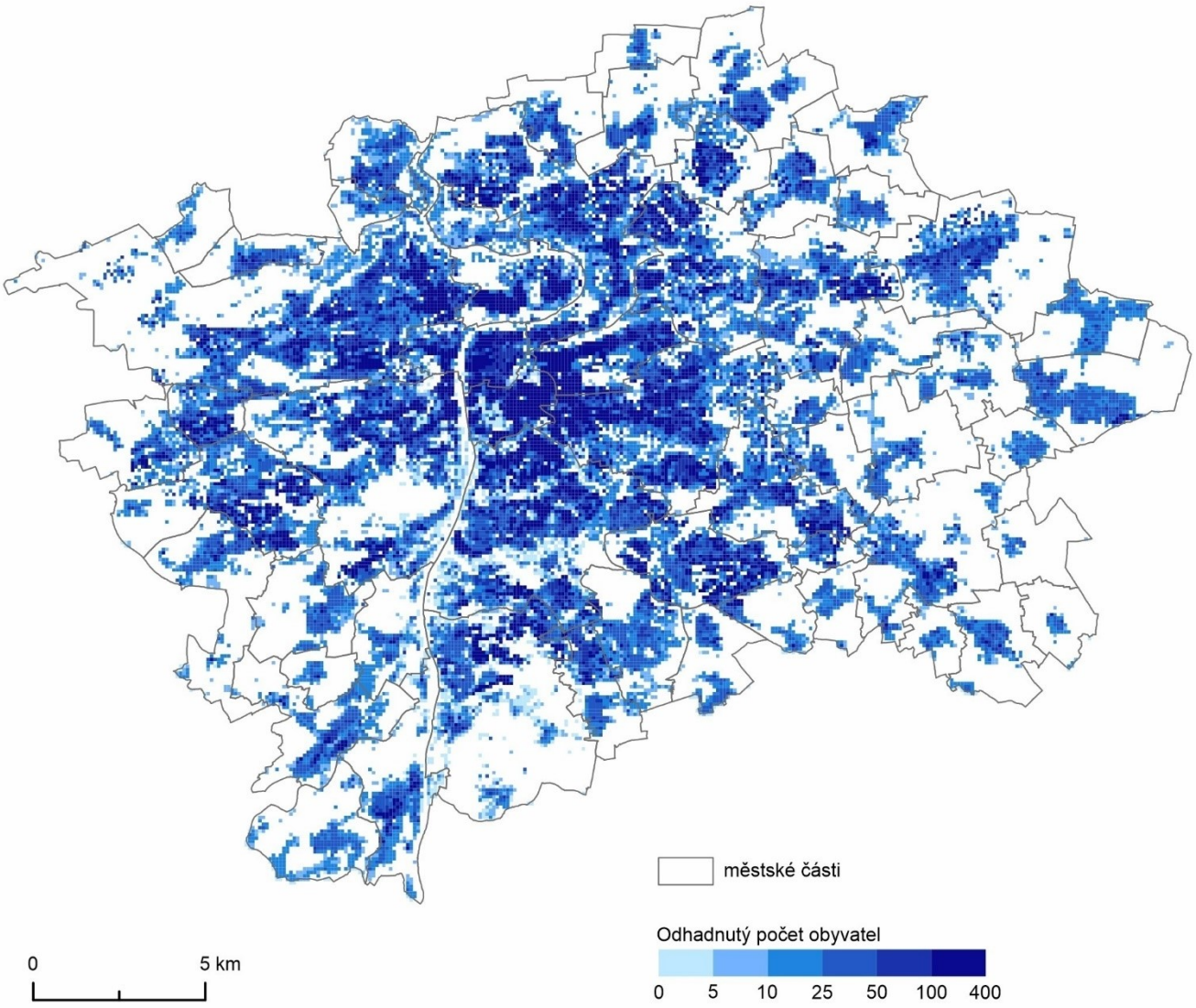
Tabulka č. 6: Výsledky desagregace nad různými velikostmi mřížky. Zdroj: vlastní tvorba

Parametry	Území	Mřížka	RTAE
2D data (41 parametrů)	A	200 m	54,37%
2D data + 3D model (42 parametrů)	A	200 m	46,42 %
2D data bez Urban Atlasu + 3D model (33 parametrů)	A	200 m	49,18 %
2D data (41 parametrů)	A	500 m	46,87 %
2D data + 3D model (42 parametrů)	A	500 m	45,73 %
2D data bez Urban Atlasu + 3D model (33 parametrů)	A	500 m	51,15 %
2D data (41 parametrů)	A	1 000 m	61,11 %
2D data + 3D model (42 parametrů)	A	1 000 m	57,27 %
2D data bez Urban Atlasu + 3D model (33 parametrů)	A	1 000 m	65,13 %

6.2.4 Hodnocení

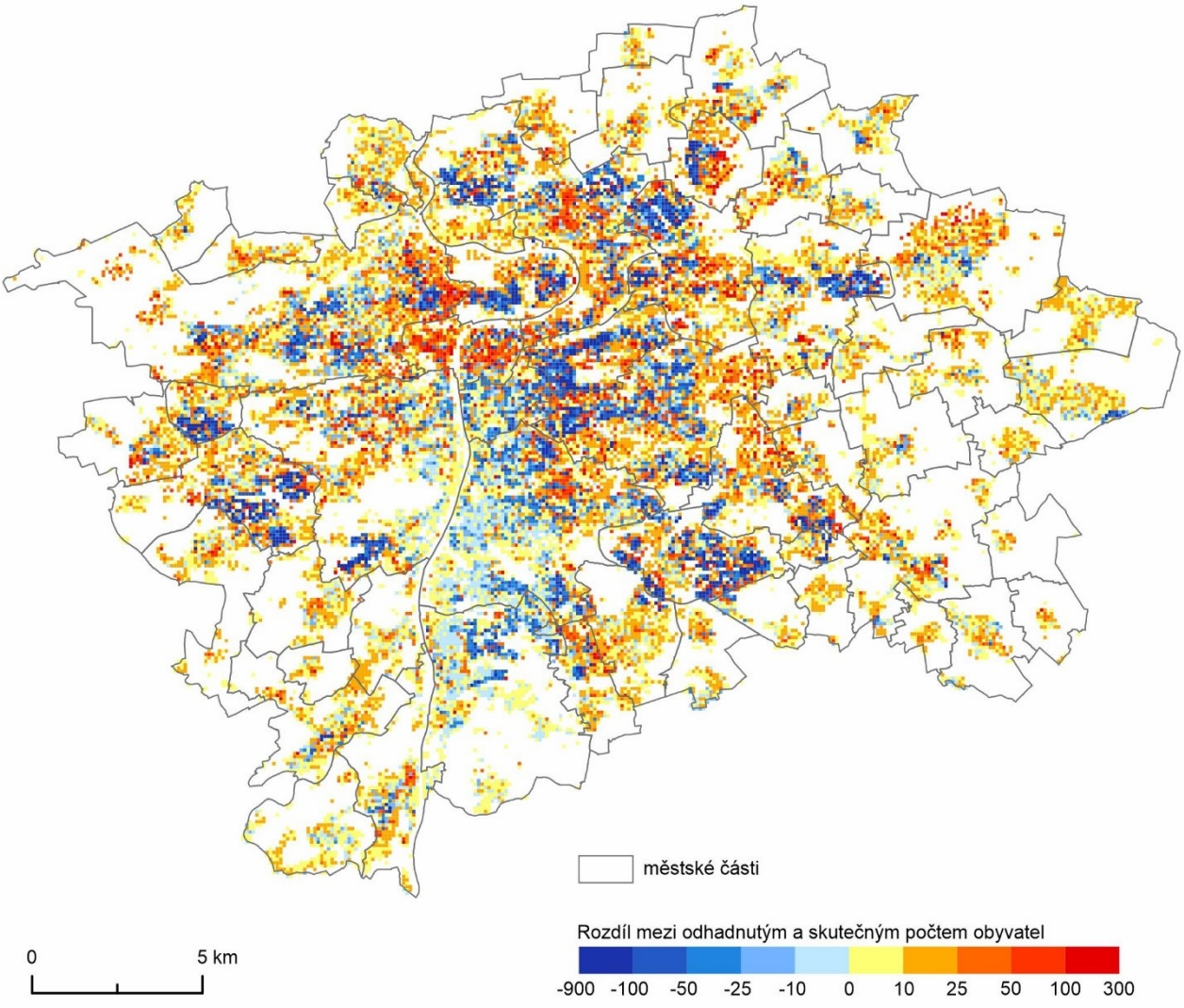
Výstup z modelu lze propojit atributem `geohash` s mřížkou, a tedy získat vizualizaci modelovaných hodnot o počtu obyvatel. Počty obyvatel nad 300 osob na čtverec mřížky byly odhadnuty zejména v oblastech sídlišť, například Lužiny, Chodov, Opatov či Černý Most. Panelový dům obecně charakterizuje relativně malá plocha podstavy, nicméně velký počet pater, z čehož můžeme usuzovat, že nově přidaná informace o objemu budovy zpřesnila odhad. Na příkladu čtverce v oblasti sídliště Nové Butovice můžeme demonstrovat určité zpřesnění, kdy skutečná hodnota obyvatel žijících v tomto čtverci činí 172 obyvatel, modelovaná hodnota s 2D daty 125 obyvatel a s 3D daty 170 obyvatel.

Obrázek č. 7: Mapa modelovaných hodnot o počtu obyvatel. Trénováno na území A s mřížkou 100x100m, zahrnuta 2D data a objem vypočtený z 3D modelu. Zdroj: vlastní tvorba



Data o skutečném počtu obyvatel za adresní bod, na kterých byl model trénován, máme k dispozici za celou Prahu. Snadno lze tedy provést porovnání skutečných a modelovaných hodnot odečtením modelovaných hodnot od hodnot skutečných.

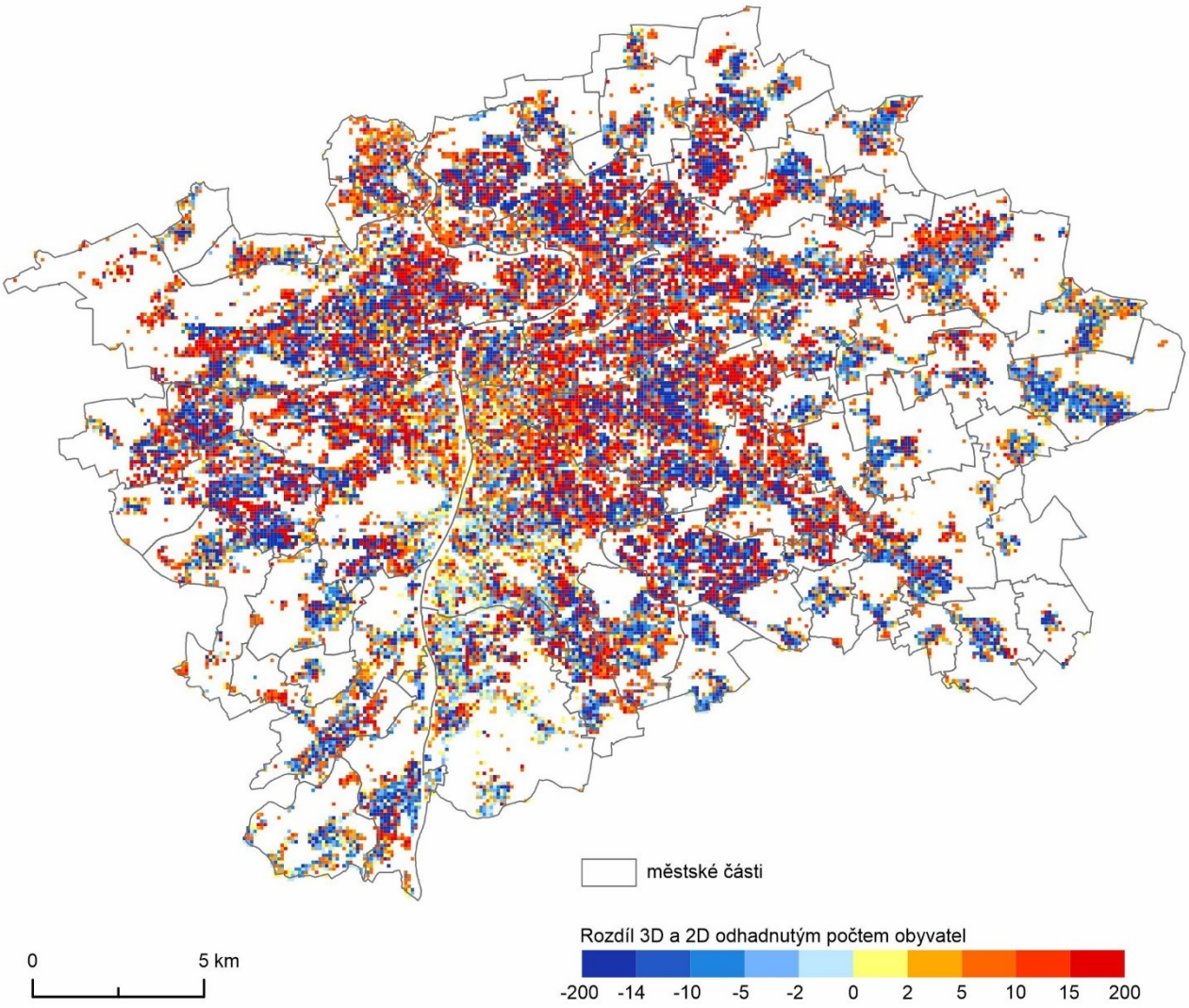
Obrázek č. 8: Mapa rozdílu modelovaných hodnot a skutečných hodnot o počtu obyvatel.
Trénováno na území A s mřížkou 100x100m, zahrnuta 2D data a objem vypočtený z 3D modelu.
Zdroj: vlastní tvorba



Z mapy (obrázek č. 7) je patrné, že nejmenších odchylek modelované hodnoty od hodnoty skutečné model dosáhl v oblasti trénovacího území. K podhodnocení počtu obyvatel došlo zejména v případě sídlišť, kde je v relativně objemově malých bytech koncentrován velký počet obyvatel. Další oblastí podhodnocení jsou části souvislé městské zástavby jako Vinohrady či Holešovice, zřejmě díky skutečnosti, že v trénovacím území modelu nebylo zahrnuto mnoho oblastí s tímto typem zástavby. Naopak nadhodnocování proběhlo zejména v oblastech s jinou, než residenční funkcí – Na Starém Městě, Malé Straně

či v oblasti velvyslanectví v Bubenči. K lokálním extrémům podhodnocení dochází v případě Janem Šimberou (2016) zmiňovaných radnic, ale i dalších objektů, jako jsou ubytovny (například ubytovna Tip ve Vysočanech – skutečný počet obyvatel 966, modelovaná hodnota s 2D daty 37 obyvatel, s 3D daty 21 obyvatel. Dalším extrémem jsou věznice, tedy Ruzyně a Pankrác. Ve Vazební věznici Ruzyně je hlášeno 734 lidí, avšak modelovaná hodnota s 2D daty vyšla 39 a s 3D daty 7 lidí.

Obrázek č. 9: Mapa rozdílu modelovaných hodnot s 3D daty a bez 3D dat. Vznikla odečtením hodnoty, modelované s 2D daty, od hodnoty modelované s 3D daty. Trénováno na území A s mřížkou 100x100m. Zdroj: vlastní tvorba



7 Závěr

Tématem a zároveň cílem této bakalářské práce byla desagregace prostorových dat za pomocí 3D modelu města Prahy. Samotné desagregaci předcházelo zpracování dat, kde byla navržena a implementována metoda výpočtu objemu budov z 3D modelu v prostředí ArcPy z rastru relativních výšek budov v prostředí ArcPy a PostreSQL s extenzí PostGIS.

V úvodu práce byla provedena rešerše principu desagregace a dasymetrického mapování. Během praktické části práce je nadále pracováno s 2D i s 3D daty, proto se rešerše literatury k dasymetrickému mapování a desagregaci zabývala oběma přístupy. Další oblastí, které se úvodní přehled věnuje, jsou datové reprezentace trojdimenzionálních objektů a na to navazující úroveň detailu z pohledu dvou rovin, geometrické a sémantické. Důležité rovněž bylo provést rešerši aplikačního rámce, tedy statistického modelu vytvořeného Janem Šimberou (2020) tak, aby navržená metoda úpravy a implementace 3D dat do statistického modelu byla kompatibilní se současnými metodami zpracování 2D dat. Popsáno je také základní schéma chodu modelu, který byl použit při desagregaci.

Všechna data, jež byla použita při statistickém modelování, popisuji v další části. Mimo 3D dat se kapitola věnuje i 2D datům, hodnoceny jsou všechny vstupy do statistického modelu z hlediska přesnosti, časového horizontu, úplnosti a dalších charakteristik tak, aby v dalších částech mohl být posuzován potenciál nahrazení 2D dat 3D daty.

Metodická část popisuje implementaci navržených metod výpočtu objemu třemi způsoby, a to sice skrze knihovnu ArcOy z 3D modelu a z rastru relativních výšek budov a skrze PostgreSQL databázi s extenzí PostGIS z relativních výšek budov. Dále je navržena metoda začlenění nově vytvořené informace o objemu budov do mřížky statistického modelu. Přiblížena jsou území, na kterých probíhalo trénování modelu a také způsob validace výstupů z modelu.

Proběhla aplikace modelu s různými kombinacemi celkem 44 vstupních parametrů. Bylo vytvořeno 70 modelů s různými vstupními parametry, nastavením modelu a typem mřížky. Na základě skutečných populačních dat proběhla validace výsledků všech modelů, což přineslo nové poznatky o využití 3D dat v rámci desagregace.

Celkové zlepšení výsledků desagregace při využití 3D dat proběhlo pouze o několik procent, nicméně ukázaly se zde dva trendy, a to rostoucí význam 3D dat při trénování na menším území a také částečné nahrazení informace, obsažené v Urban Atlasu, právě 3D daty. Při případné studii na území, kde Urban Atlas není dostupný mohou 3D data posloužit jako náhrada, která v některých případech může vést i k lepším výsledkům desagregace. Vyšší výkon modelu při trénování na menším území za použití 3D dat má také určitý potenciál v praxi, pokud by byl model například trénován na datech, dostupných pouze pro malou část zájmového území.

V rámci práce byla vyvinuta a implementována metoda výpočtu objemu z 3D modelu a relativních výšek budov a metoda aplikace informace do modelu Jana Šimbery (2020) tak, aby byl vstup kompatibilní s ostatními vstupy. Proběhlo modelování s různými kombinacemi parametrů, které přineslo nové poznatky o využitelnosti 3D dat v desagregaci prostorových dat.

Pokud by byla vyřešena vysoká výpočetní náročnost a dostupnost lidarových dat, mohla by 3D data posloužit k vylepšení populačních mřížek, podobně jak činí Rosina et al. (2017) v případě OSM dat. Zajímavým pokračováním či doplněním této práce by rovněž bylo desagregovat data pro jiné město užitím modelu, jenž by byl trénován na celé Praze, případně prozkoumat vliv, jaký má výšková rozmanitost města na přínos využití 3D modelu během desagregace. V případě města, které zahrnuje výškové budovy v centru a rodinné domy na okrajích by měl být přínos 3D dat během desagregace vyšší než v případě Prahy, která je výškově relativně sourodá.

Mimo zpracování a využití 3D dat ovšem uvažujme i možný vývoj a rozšíření statistického modelování. Zajímavým rozšířením by byla implementace dalších způsobů strojového učení, jako jsou umělé neuronové sítě. Model použitý v této práci desagreguje populační hodnoty v jednom určitém časovém horizontu, který závisí na vstupních datech. Prediktivní modelování, které by z dat za minulé časové období, případně se zcela jiným přístupem, například z územních plánů odhadovalo budoucí vývoj populačních charakteristik by mohlo být velmi užitečné při porozumění například suburbanizačním

procesům či gentrifikaci a mohlo by napomoci porozumění těmto lokálním sociálně-kulturním změnám.

8 Literatura a zdroje

ANDERSON, S. J., TUTTLE, B. T., POWELL, R.L., SUTTON, P.C. (2010) Characterizing relationships between population density and nighttime imagery for Denver, Colorado: issues of scale and representation. International Journal of Remote Sensing, 31, 21, 5733–5746.

ARCDATA (2020): ArcČR 500 - digitální geografická databáze, verze 3.3. https://www.arcdata.cz/produkty/geograficka-data/arccr-500 (cit. 20. 10. 2019).

BAKILLAH, M., LIANG, S., MOBASHERI, A., JOKAR ARSANJANI, J., & ZIPF, A. (2014): Fine-resolution population mapping using OpenStreetMap points-of-interest. International Journal of Geographical Information Science, 28, 9, 1940–1963.

BILJECKI, F., OHORI, A. K., LEDOUX, H., PETERS, R., STOTER, J. (2016): Population Estimation Using a 3D City Model: A Multi Scale Country Wide Study in the Netherlands. PLoS ONE, 11, 6.

CGAL (2020): The Computational Geometry Algorithms Library. https://www.cgal.org (cit. 25. 04. 2020).

DMOWSKA, A., STEPINSKI, T. F. (2017): A high resolution population grid for the conterminous United States: The 2010 edition. Computers Enviroment and Urban Systems, 61, 1, 13–23.

DONG, P., RAMESH, S., NEPALI, A. (2010): Evaluation of small-area population estimation using LiDAR, Landsat TM and parcel data. International Journal of Remote Sensing, 31, 21, 5571–5586.

ESRI (2020a): Fundamentals of 3D data, https://desktop.arcgis.com/en/arcmap/latest/extensions/3d-analyst/fundamentals-of-3d-data.htm (cit. 14. 3. 2020).

ESRI (2020b): The Multipatch Geometry Type, https://support.esri.com/en/white-paper/1483?rmedium=whitepaper-product-by-metaid (cit. 14. 3. 2020).

EVROPSKÁ KOMISE (2012): Urban Atlas 2012. https://land.copernicus.eu/local/urban-atlas/urban-atlas-2012 (cit. 12.1.2020).

EVROPSKÁ KOMISE (2016): Mapping Guide for a European Urban Atlas. https://land.copernicus.eu/user-corner/technical-library/urban-atlas-mapping-guide (cit. 5.3. 2020).

GEOFABRIK (2020): OSM download server. https://download.geofabrik.de (cit. 20. 1. 2020).

HERRING, J., R. (2011): OpenGIS® Implementation Standard for Geographic information – Simple feature access – Part 1: Common architecture, Open Geospatial Consortium, https://www.ogc.org/standards/sfa (cit. 15. 3. 2020).

IPR (2017): Relativní výšky budov. http://opendata.praha.eu/dataset/ipr-relativni_vysky_budov (cit. 25. 10. 2019).

IPR (2019a): Budovy3D. http://opendata.praha.eu/dataset/ipr-budovy_3d (cit. 25. 10. 2019).

IPR (2019b): Podlažnosti. http://opendata.praha.eu/dataset/ipr-podlaznosti (cit. 25. 10. 2019).

IPR (2019c): Městské části. http://opendata.praha.eu/dataset/ipr-mestske_casti (cit. 20. 4. 2020).

IPR (2020): Digitální technická mapa Prahy – plochy (polygony) budov. https://www.geoportalpraha.cz/cs/data/otevrena-data/C170F739-D27C-4556-9138-CAF7C14FB01B (cit. 11. 2. 2020).

LANGFORD, M., (2006): Obtaining population estimates in non-census reporting zones: an evaluation of the 3-class dasymetric method. Computers, Environment and Urban Systems, 30, 2, 161–180.

LANGFORD, M., HIGGS, G., RADCLIFFE, STEPHEN, J., WHITE, J., DAMIAN, S. (2008): Urban population distribution models and service accessibility estimation. Computers, Enviroment and Urban Systems, 32, 1, 66–80.

LI, X., ZHOU, W. (2018): Dasymetric mapping of urban population in China based on radiance corrected DMSP-OLS nighttime light and land cover data. Science of The Total Enviroment, 643, 1, 1248–1256.

LU, Z., IM, J., QUACKENBUSH, L. (2011): A Volumetric Approach to Population Estimation Using Lidar Remote Sensing. Photogrammetric Engineering and Remote Sensing, 77, 11, 1145–1156.

MAANTAY, J.A., MAROKO, A.R., HERRMANN, C., (2007): Mapping population distribution in the urban environment: the cadastral-based expert dasymetric system (CEDS). Cartography and Geographic Information Science, 34, 2, 77–102.

POSTGIS (2020): Spatial and Geographic objects for PostgreSQL. https://postgis.net (cit. 12. 02. 2020).

POZZI, F., SMALL, C. (2005): Analysis of urban land cover and population density in the United States. Photogrammetric Engineering and Remote Sensing, 71, 6, 719–726.

ROSINA, K., HURBÁNEK, P., CEBECAUER, M. (2017): Using OpenStreetMap to improve population grids in Europe. Cartography and Geographic Information Science, 44, 2, 139–151.

SADAHIRO, Y. (1999): Accuracy of areal interpolation: A comparison of alternative methods. Journal of Geographical Systems, 1, 4, 323–346.

SHUO-SHENG, W., LE, W., XIAOMIN, Q. (2008): Incorporating GIS Building Data and Census Housing Statistics for Sub-Block-Level Population Estimation. The Professional Geographer, 60, 1, 121–135.

SLDB (2011): Budovy s číslem domovním a vchody (statistické budovy) - bod. Český statistický úřad, Praha (cit. 10. 1. 2020).

STEIGER, E., WESTERHOLT, R., RESCH, B., ZIPF, A. (2015): Twitter as an indicator for whereabouts of people? Correlating Twitter with UK census data. Computers, Environment and Urban Systems, 54, 255–265.

SUTTON, P. (1997): Modeling population density with night-time satellite imagery and GIS. Computers, Environment and Urban Systems, 21, 3–4, 227–244.

ŠIMBERA, J. (2016): Modelování charakteristik obyvatelstva z topografických dat. Diplomová práce. Katedra aplikované geoinformatiky a kartografie PřF UK, Praha.

ŠIMBERA, J. (2020): Neighborhood features in geospatial machine learning: the case of population disaggregation. Cartography and Geographic Information Science, 47, 1, 79–94.

USGS (2020): EarthExplorer. https://earthexplorer.usgs.gov (cit. 5. 3. 2020).

WANG, S., TIAN, Y., ZHOU, Y., LIU, W., LIN, CH. (2016): Fine-Scale Population Estimation by 3D Reconstruction of Urban Residential Buildings. Sensors, 16, 10.

WU, S., LE, W., QIU, X. (2008): Incorporating GIS Building Data and Census Housing Statistics for Sub-Block-Level Population Estimation. The Professional Geographer, 60, 1, 121–135.

ZASINA, J. (2018): The Instagram Image of the City. Insights from Lodz, Poland. Bulletin of Geography. Socio-economic Series, 42, 42, 213–225.

ŽÁRA, J., BENEŠ, B., SOCHOR, J., FELKEL, P. (2005): Moderní počítačová grafika. Computer Press, Praha.